15BEEC_E29 DIGITAL IMAGE PROCESSING 3 0 0 3 100

OBJECTIVES

- To understand the Fundamentals of image processing.
- To learn Various transforms used in image processing.
- To learn the Image processing techniques like image enhancement, reconstruction, compression and segmentation.

INTENDED OUTCOMES:

- Understand the Fundamentals of image processing.
- Knowledge about various transforms used in image processing.
- Knowledge about the Image processing techniques like image enhancement, reconstruction, compression and segmentation.

UNIT I-DIGITAL IMAGE FUNDAMENTALS

Introduction-Elements of Digital Image Processing system- elements of visual perception – image sensing and acquisition – Image sampling and quantization - image representation -Some basic relationship between pixels.

UNIT II-IMAGE TRANSFORMS

Introduction - 2D Discrete Fourier Transform – Properties- Importance of Phase -Walsh – Hadamard – Discrete Cosine Transform, Haar, –KL transforms –Singular Value Decomposition.

UNIT III-IMAGE ENHANCEMENT

Enhancement through point operation- Histogram manipulation – Gray level transformation-Neighbourhood operation – Median filter - Image Sharpening- Bit plane slicing - Homomorphic Filtering – Zooming operation.

UNIT IV-IMAGE RESTORATION

Model of Image Degradation/restoration process –Inverse filtering -Least mean square (Wiener) filtering – Constrained least mean square restoration – Singular value decomposition-Recursive filtering.

UNIT V-IMAGE COMPRESSION AND SEGMENTATION

Image compression schemes – Information theory – Run length, Huffman and arithmetic coding – Vector quantization - JPEG. Image Segmentation – Classification – Thresholding – edge based segmentation – Hough transform – Active contour.

EXT BOOKS :

S.NO.	Author(s) Name	Title of the book	Publisher	Year of the publication
1.	Rafael C Gonzalez and Richard E Woods,	Digital Image Processing	Pearson Education, 3rd Edition	**2003**
2.	S. Jayarman, S. Esakkirajan and T. Veerakumar,	Digital Image Processing	Tata McGraw Hill	**2010**

REFERENCES:

S.NO.	Author(s) Name	Title of the book	Publisher	Year of the publication
1.	William K Pratt	Digital Image Processing	John Willey	**2001**
2.	Millman Sonka, Vaclav Hlavac, Roger Boyle, and Broos Colic	Image Processing Analysis and Machine Vision	Thompson learning	1999
3.	A.K. Jain	Fundamentals of Digital Image Processing	Pearson Education	**1989**

UNIT-1

DIGITAL IMAGE FUNDAMENTALS AND TRANSFORMS INTRODUCTION and FUNDAMENTALS

1.1 INTRODUCTION

The digital image processing deals with developing a digital system that performs operations on a digital image.
An image is nothing more than a two dimensional signal. It is defined by the mathematical function f(x,y) where x and y are the two co-ordinates horizontally and vertically and the amplitude of f at any pair of coordinate (x, y) is called the intensity or gray level of the image at that point.
When x, y and the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of image digital image processing refers to the processing of digital image by means of a digital computer.
A digital image is composed of a finite number of elements, each of which has a particular location and values of these elements are referred to as picture elements, image elements, pels and pixels.

1.1.1 Motivation and Perspective

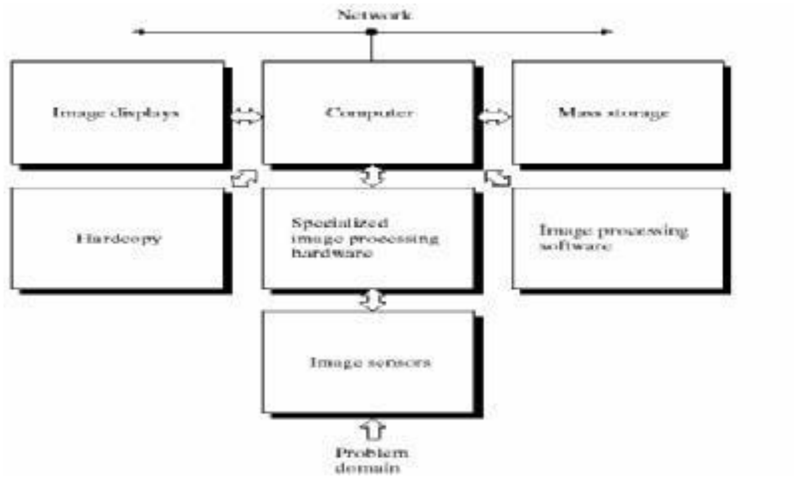
Digital image processing deals with manipulation of digital images through a digital computer. It is a subfield of signals and systems but focus particularly on images. DIP focuses on developing a computer system that is able to perform processing on an image. The input of that system is a digital image and the system process that image using efficient algorithms, and gives an image as an output. The most common example is Adobe Photoshop. It is one of the widely used application for processing digital images.

1.1.2 Applications

Some of the major fields in which digital image processing is widely used are mentioned below

(1) Gamma Ray Imaging- Nuclear medicine and astronomical observations.
(2) X-Ray imaging – X-rays of body.
(3) Ultraviolet Band –Lithography, industrial inspection, microscopy, lasers.
(4) Visual And Infrared Band – Remote sensing.
(5) Microwave Band – Radar imaging.

1.1.3 **<u>Components of Image Processing System</u>**



i) <u>Image Sensors</u>

With reference to sensing, two elements are required to acquire digital image. The first is a physical device that is sensitive to the energy radiated by the object we wish to image and second is specialized image processing hardware.

ii) <u>Specialize image processing hardware</u> –

It consists of the digitizer just mentioned, plus hardware that performs other primitive operations such as an arithmetic logic unit, which performs arithmetic such addition and subtraction and logical operations in parallel on images

iii) <u>Computer</u>

It is a general purpose computer and can range from a PC to a supercomputer depending on the application. In dedicated applications, sometimes specially designed computer are used to achieve a required level of performance

iv) <u>Software</u>

It consist of specialized modules that perform specific tasks a well designed package also includes capability for the user to write code, as a minimum, utilizes the specialized module. More sophisticated software packages allow the integration of these modules.

v) <u>Mass storage –</u>

This capability is a must in image processing applications. An image of size 1024 x1024 pixels ,in which the intensity of each pixel is an 8- bit quantity requires one megabytes of storage space if the image is not compressed .Image processing applications falls into three principal categories of storage

i) Short term storage for use during processing
ii) On line storage for relatively fast retrieval
iii) Archival storage such as magnetic tapes and disks

vi) Image displays-

Image displays in use today are mainly color TV monitors. These monitors are driven by the outputs of image and graphics displays cards that are an integral part of computer system

vii) Hardcopy devices -

The devices for recording image includes laser printers, film cameras, heat sensitive devices inkjet units and digital units such as optical and CD ROM disk. Films provide the highest possible resolution, but paper is the obvious medium of choice for written applications.

viii) Networking

It is almost a default function in any computer system in use today because of the large amount of data inherent in image processing applications. The key consideration in image transmission bandwidth.

1.1.4 Elements of Visual Perception

1.1.4.1 Structure of the human Eye

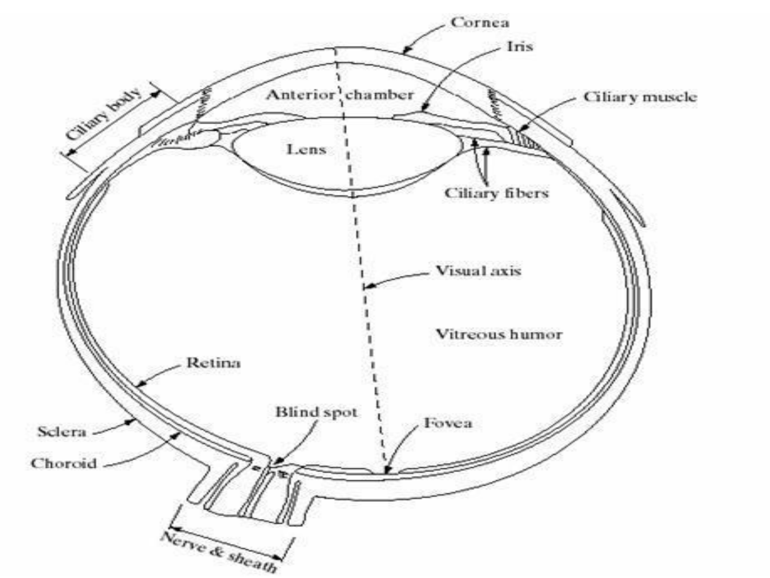
The eye is nearly a sphere with average approximately 20 mm diameter. The eye is enclosed with three membranes

a) The cornea and sclera - it is a tough, transparent tissue that covers the anterior surface of the eye. Rest of the optic globe is covered by the sclera

b) The choroid –
It contains a network of blood vessels that serve as the major source of nutrition to the eyes. It helps to reduce extraneous light entering in the eye It has two parts

(1) Iris Diaphragms- it contracts or expands to control the amount of light that enters the eyes

(2) Ciliary body



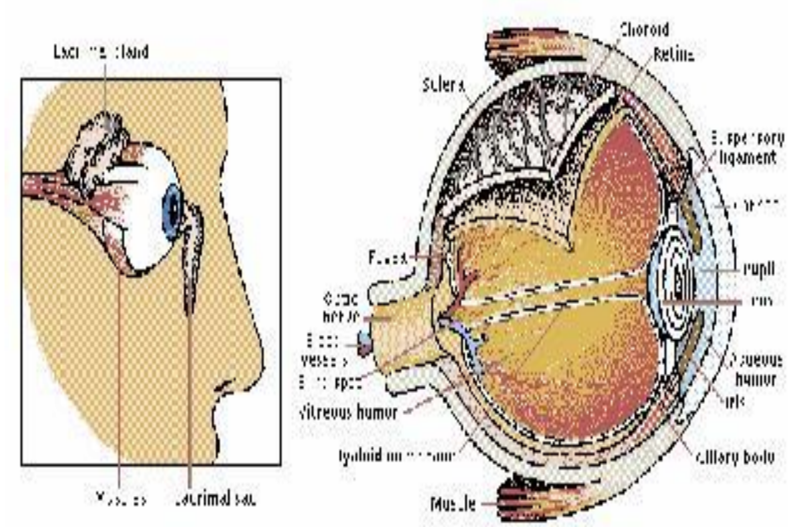
(c) Retina – it is innermost membrane of the eye. When the eye is properly focused, light from an object outside the eye is imaged on the retina. There are various light receptors over the surface of the retina
The two major classes of the receptors are-

1) cones- it is in the number about 6 to 7 million. These are located in the central portion of the retina called the fovea. These are highly sensitive to

color. Human can resolve fine details with these cones because each one is connected to its own nerve end. Cone vision is called photopic or bright light vision

2) Rods – these are very much in number from 75 to 150 million and are distributed over the entire retinal surface. The large area of distribution and the fact that several roads are connected to a single nerve give a general overall picture of the field of view. They are not involved in the color vision and are sensitive to low level of illumination. Rod vision is called is scotopic or dim light vision.

The absent of reciprocators is called blind spot



1.1.4.2 Image Formation in the Eye

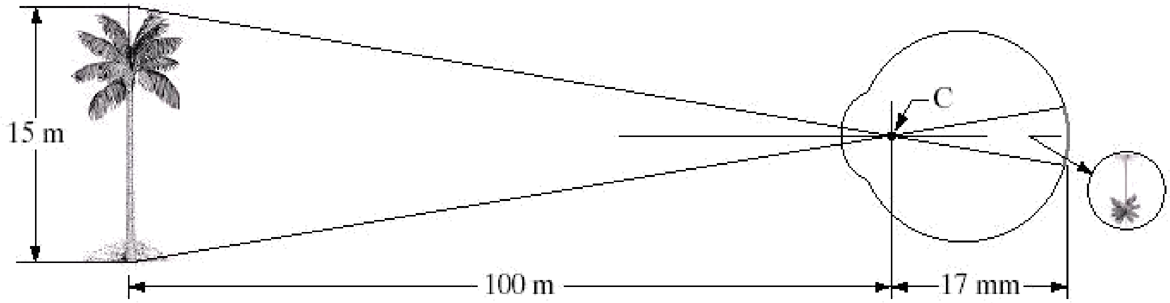
The major difference between the lens of the eye and an ordinary optical lens in that the former is flexible.

The shape of the lens of the eye is controlled by tension in the fiber of the ciliary body. To focus on the distant object the controlling muscles allow the lens to become thicker in order to focus on object near the eye it becomes relatively flattened.

The distance between the center of the lens and the retina is called the focal length and it varies from 17mm to 14mm as the refractive power of the lens increases from its minimum to its maximum**.**

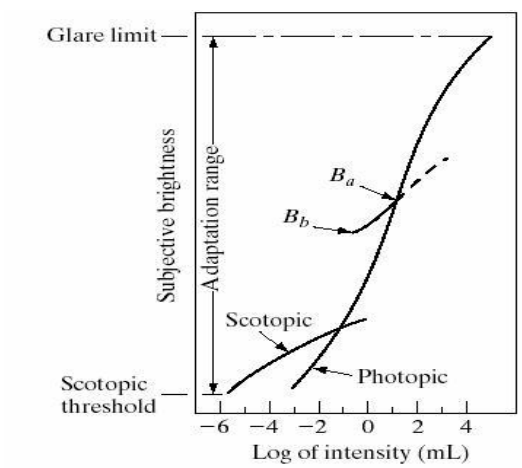
When the eye focuses on an object farther away than about 3m.the lens exhibits its lowest refractive power. When the eye focuses on a nearly object. The lens is most strongly refractive.

The retinal image is reflected primarily in the area of the fovea. Perception then takes place by the relative excitation of light receptors, which transform radiant energy into electrical impulses that are ultimately decoded by the brain.



1.1.4.3 Brightness Adaption and Discrimination

Digital image are displayed as a discrete set of intensities. The range of light intensity levels to which the human visual system can adopt is enormous- on the order of 10^{10}- from scotopic threshold to the glare limit. Experimental evidences indicate that subjective brightness is a logarithmic function of the light intensity incident on the eye.



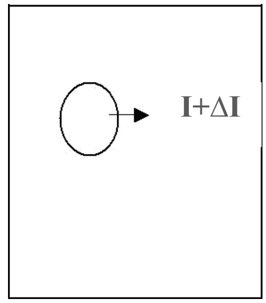
The curve represents the range of intensities to which the visual system can adopt. But the visual system cannot operate over such a dynamic range simultaneously. Rather, it is accomplished by change in its overcall sensitivity called brightness adaptation.

For any given set of conditions, the current sensitivity level to which of the visual system is called brightness adoption level , B_a in the curve. The small intersecting curve represents the range of subjective brightness that the eye can perceive when adapted to this level. It is restricted at level B_b , at and below which all stimuli are perceived as indistinguishable blacks. The upper portion of the curve is not actually restricted. whole simply raise the adaptation level higher than B_a .

The ability of the eye to discriminate between change in light intensity at any specific adaptation level is also of considerable interest.

Take a flat, uniformly illuminated area large enough to occupy the entire field of view of the subject. It may be a diffuser such as an opaque glass, that is illuminated from behind by a light source whose intensity, I can be varied. To this field is added an increment of illumination ΔI in the form of a short duration flash that appears as circle in the center of the uniformly illuminated field.

If ΔI is not bright enough, the subject cannot see any perceivable changes.



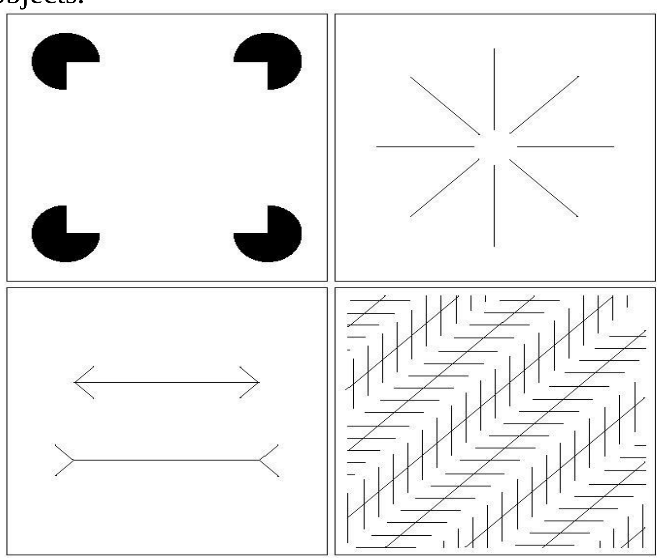
As ΔI gets stronger the subject may indicate of a perceived change. ΔI_c is the increment of illumination discernible 50% of the time with background illumination I. Now, ΔI_c /I is called the Weber ratio.

Small value means that small percentage change in intensity is discernible representing “good” brightness discrimination.

Large value of Weber ratio means large percentage change in intensity is required representing “poor brightness discrimination”.

1.1.4.4 Optical illusion

In this the eye fills the non existing information or wrongly pervious geometrical properties of objects.



1.1.5 **Fundamental Steps in Digital Image Processing**

There are two categories of the steps involved in the image processing –

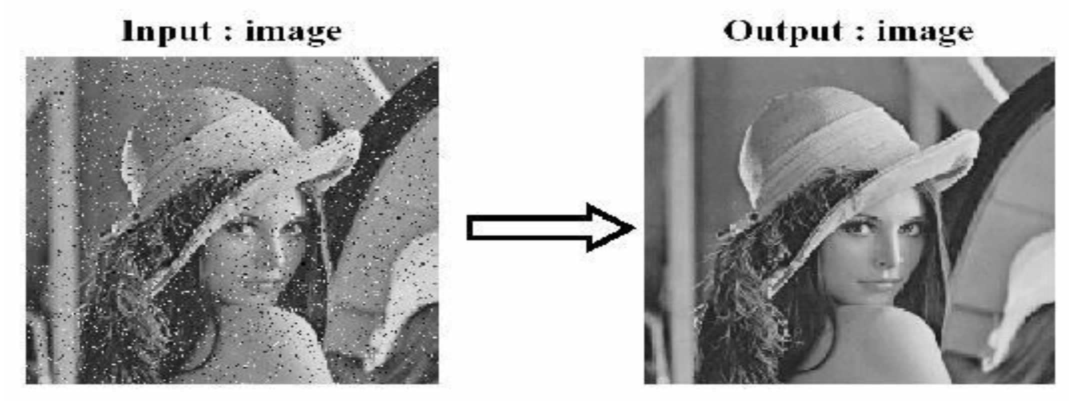
(1) Methods whose outputs are input are images.
(2) Methods whose outputs are attributes extracted from those images.

i) Image acquisition

It could be as simple as being given an image that is already in digital form. Generally the image acquisition stage involves processing such scaling.

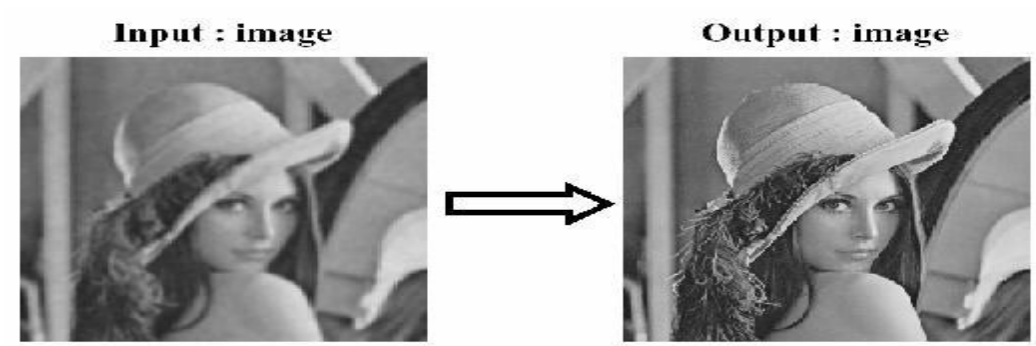
ii) Image Enhancement

It is among the simplest and most appealing areas of digital image processing. The idea behind this is to bring out details that are obscured or simply to highlight certain features of interest in image. Image enhancement is a very subjective area of image processing.



iii) <u>Image Restoration</u> –

It deals with improving the appearance of an image. It is an objective approach, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image processing. Enhancement, on the other hand is based on human subjective preferences regarding what constitutes a “good” enhancement result



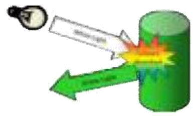
<table>
<tr><td>Color Image Processing</td><td>Wavelets & Multiresolution Processing</td><td>Image Compression</td><td>Morphological Image Processing</td></tr>
<tr><td>Image Restoration</td><td colspan="2" rowspan="3">Knowledge Base</td><td>Image Segmentation</td></tr>
<tr><td>Image Enhancement</td><td>Representation and description</td></tr>
<tr><td>Image Acquisition</td><td>Objects recognition</td></tr>
</table>

Fig: Fundamental Steps in DIP

.

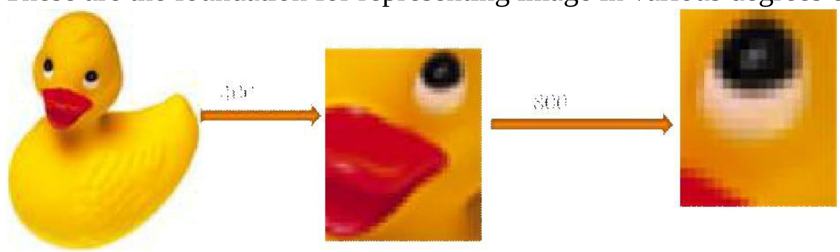
iv) <u>Color image processing</u> –

It is an area that is been gaining importance because of the use of digital images over the internet. Color image processing deals with basically color models and their implementation in image processing applications.



v) Wavelets and Multiresolution Processing -

These are the foundation for representing image in various degrees of resolution

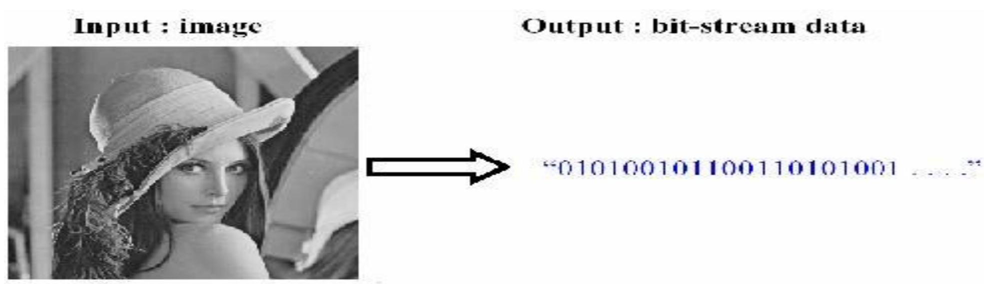


vi) Compression -

It deals with techniques reducing the storage required to save an image, or the bandwidth required to transmit it over the network. It has to major approaches

a) Lossless Compression

b) Lossy Compression



vii) Morphological processing –

It deals with tools for extracting image components that are useful in the representation and description of shape and boundary of objects. It is majorly used in automated inspection applications.

viii) Representation and Description-

It always follows the output of segmentation step that is, raw pixel data, constituting either the boundary of an image or points in the region itself. In either case converting the data to a form suitable for computer processing is necessary.

ix) Recognition –

It is the process that assigns label to an object based on its descriptors. It is the last step of image processing which use artificial intelligence of softwares.

Knowledge base

Knowledge about a problem domain is coded into an image processing system in the form of a knowledge base. This knowledge may be as simple as detailing regions of an image where the information of the interest in known to be located. Thus limiting search that has to be conducted in seeking the information. The knowledge base also can be quite complex such interrelated list of all major possible defects in a materials inspection problems or an image database containing high resolution satellite images of a region in connection with change detection application

1.1.6 <u>**A Simple Image Model**</u>

An image is denoted by a two dimensional function of the form f{x, y}. The value or amplitude of f at spatial coordinates {x,y} is a positive scalar quantity whose physical meaning is determined by the source of the image.
When an image is generated by a physical process, its values are proportional to energy radiated by a physical source. As a consequence, f(x,y) must be nonzero and finite; that is

$$o<f(x,y) <co$$

The function f(x,y) may be characterized by two components-

The amount of the source illumination incident on the scene being viewed.
The amount of the source illumination reflected back by the objects in the scene

These are called illumination and reflectance components and are denoted by i (x,y) an r (x,y) respectively.
The functions combine as a product to form f(x,y)
We call the intensity of a monochrome image at any coordinates (x,y) the gray level (l) of the image at that point

$$l= f (x, y.)$$
$$L\ min \leq l \leq Lmax$$

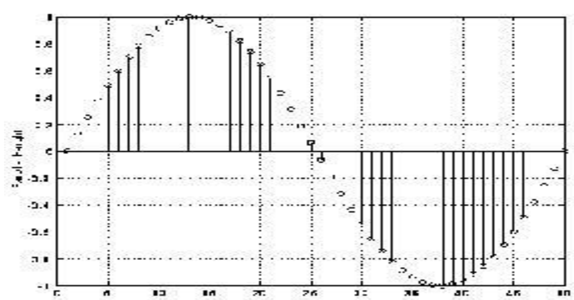
Lmin is to be positive and Lmax must be finite

$$Lmin = imin\ rmin$$
$$Lamx = imax\ rmax$$

The interval [Lmin, Lmax] is called gray scale. Common practice is to shift this interval numerically to the interval [0, L-l] where l=0 is considered black and l= L-1 is considered white on the gray scale. All intermediate values are shades of gray of gray varying from black to white.

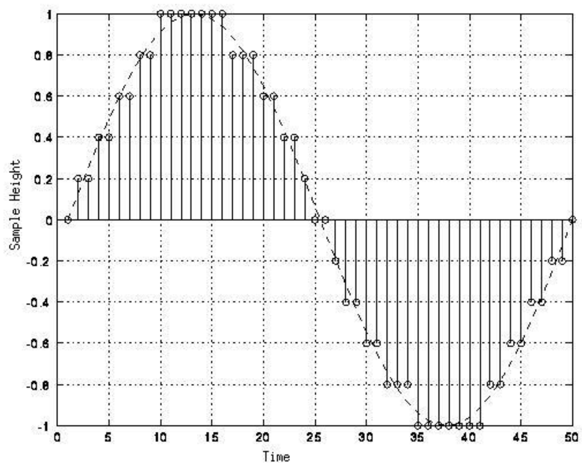
1.1.7 <u>**Image Sampling And Quantization**</u>

To create a digital image, we need to convert the continuous sensed data into digital from. This involves two processes – sampling and quantization. An image may be continuous with respect to the x and y coordinates and also in amplitude. To convert it into digital form we have to sample the function in both coordinates and in amplitudes.
Digitalizing the coordinate values is called sampling
Digitalizing the amplitude values is called quantization
There is a continuous the image along the line segment AB.
To simple this function, we take equally spaced samples along line AB. The location of each samples is given by a vertical tick back (mark) in the bottom part. The samples are shown as block squares superimposed on function the set of these discrete locations gives the sampled function.



In order to form a digital, the gray level values must also be converted (quantized) into discrete quantities. So we divide the gray level scale into eight discrete levels ranging from block to white. The vertical tick mark assign the specific value assigned to each of the eight level values.

The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment it made depending on the vertical proximity of a simple to a vertical tick mark.

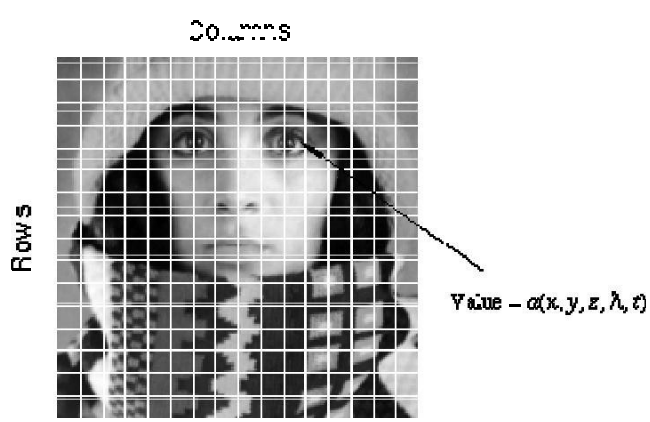


Starting at the top of the image and covering out this procedure line by line produces a two dimensional digital image.

1.1.8 **Digital Image Definition**

A digital image $a[m,n]$ described in a 2D discrete space is derived from an analog image $a(x,y)$ in a 2D continuous space through a sampling process that is frequently referred to as digitization. The mathematics of that sampling process will be described in subsequent Chapters. For now we will look at some basic definitions associated with the digital image. The effect of digitization is shown in figure 1.

The 2D continuous image $a(x,y)$ is divided into N rows and M columns. The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates $[m,n]$ with $\{m = 0,1,2,...,M-1\}$ and $\{n = 0,1,2,...,N-1\}$ is $f[m,n]$. In fact, in most cases $f(x,y)$, is actually a function of many variables including depth (z), color (λ) and time (t).



There are three types of computerized processes in the processing of image

1) Low level process -these involve primitive operations such as image processing to reduce noise, contrast enhancement and image sharpening. These kind of processes are characterized by fact the both inputs and output are images.

2) Mid level image processing - it involves tasks like segmentation, description of those objects to reduce them to a form suitable for computer processing, and classification of individual objects. The inputs to the process are generally images but outputs are attributes extracted from images.

3) High level processing – It involves “making sense” of an ensemble of recognized objects, as in image analysis, and performing the cognitive functions normally associated with vision.

1.1.9 **Representing Digital Images**

The result of sampling and quantization is matrix of real numbers. Assume that an image f(x,y) is sampled so that the resulting digital image has M rows and N Columns. The values of the coordinates (x,y) now become discrete quantities thus the value of the coordinates at orgin become 9X,y) =(o,o) The next Coordinates value along the first signify the iamge along the first row. it does not mean that these are the actual values of physical coordinates when the image was sampled.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,M-1) \\ f(1,0) & f(1,1) & \cdots & f(1,M-1) \\ \vdots & & & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix}$$

Thus the right side of the matrix represents a digital element, pixel or pel. The matrix can be represented in the following form as well.

The sampling process may be viewed as partitioning the xy plane into a grid with the coordinates of the center of each grid being a pair of elements from the Cartesian products Z2 which is the set of all ordered pair of elements (Zi, Zj) with Zi and Zj being integers from Z.

Hence f(x,y) is a digital image if gray level (that is, a real number from the set of real number R) to each distinct pair of coordinates (x,y). This functional assignment is the quantization process. If the gray levels are also integers, Z replaces R, the and a digital image become a 2D function whose coordinates and she amplitude value are integers.
Due to processing storage and hardware consideration, the number gray levels typically is an integer power of 2.

$$L=2^K$$

Then, the number, b, of bites required to store a digital image

is $B=M*N*k$

When M=N

The equation become $b=N^2*k$

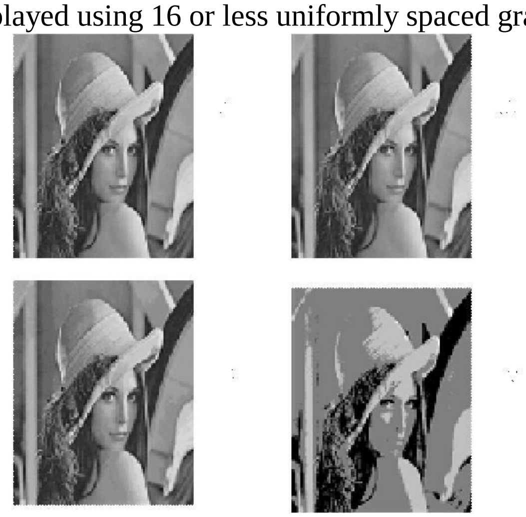
When an image can have 2^k gray levels, it is referred to as “k- bit” . An image with 256 possible gray levels is called an “8- bit image”($256=2^8$)

1.1.10 **Spatial and Gray Level Resolution**

Spatial resolution is the smallest discernible details are an image. Suppose a chart can be constructed with vertical lines of width w with the space between the also having width W, so a line pair consists of one such line and its adjacent space thus. The width of the line pair is 2w and there is 1/2w line pair per unit distance resolution is simply the smallest number of discernible line pair unit distance.

Gray levels resolution refers to smallest discernible change in gray levels

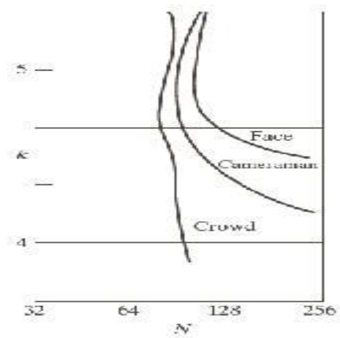
Measuring discernible change in gray levels is a highly subjective process reducing the number of bits R while repairing the spatial resolution constant creates the problem of false contouring .it is caused by the use of an insufficient number of gray levels on the smooth areas of the digital image . It is called so because the rides resemble top graphics contours in a map. It is generally quite visible in image displayed using 16 or less uniformly spaced gray levels.



1.1.11 **Iso Preference Curves**

To see the effect of varying N and R simultaneously. There picture are taken having littlie, mid level and high level of details.

Different image were generated by varying N and k and observers were then asked to rank the results according to their subjective quality. Results were summarized in the form of iso preference curve in the N-k plane.



The iospreference curve tends to shift right and upward but their shapes in each of the three image categories are shown in the figure. A shift up and right in the curve simply means large values for N and k which implies better picture quality

The result shows that iosreference curve tends to become more vertical as the detail in the image increases. The result suggests that for image with a large amount of details only a few gray levels may be needed. For a fixed value of N, the perceived quality for this type of image is nearly independent of the number of gray levels used.

1.1.12 **Zooming and Shrinking of Digital Images**

Zooming may be said oversampling and shirking may be called as under sampling these techniques are applied to a digital image.
These are two steps of zooming-

i) Creation of new pixel locations
ii) Assignment of gray level to those new locations.

⇨ In order to perform gray –level assignment for any point in the overly, we look for the closet pixel in the original image and assign its gray level to the new pixel in the grid. This method rowan as nearest neighbor interpolation

⇨ Pixel replication - Is a special case of nearest neighbor interpolation, It is applicable if we want to increase the size of an image an integer number of times.

⇨ For eg.- to increase the size of image as double. We can duplicate each column. This doubles the size of the image horizontal direction. To increase assignment of each of each vertical direction we can duplicate each row. The gray level assignment of each pixel is determined by the fact that new location are exact duplicates of old locations.

Drawbacks

(i) Although nearest neighbor interpolation is fast ,it has the undesirable feature that it produces a check board that Is not desirable

⇨ Bilinear interpolation-

Using the four nearest neighbor of a point .let (x,y) denote the coordinate of a point in the zoomed image and let v(x1,y1) denote the gray levels assigned to it .for bilinear interpolation .the assigned gray levels is given by

V(x1,y1)-ax1+by1+cx1y1+d

Where the four coefficient are determined from the four equation in four unknowns that can be writing using the four nearest neighbor of point (x1,y1)

Shrinking is done in the similar manner .the equivalent process of the pixel replication is row –column deletion .shrinking leads to the problem of aliasing.

1.1.13 **Pixel Relationships**

1.1.13.1 Neighbor of a pixel

A pixel p at coordinate (x,y) has four horizontal and vertical neighbor whose coordinate can be given by

(x+1, y) (X-1,y) (X ,y + 1) (X, y-1)

This set of pixel called the 4-neighbours
Of ,p is denoted by n4(p) ,Each pixel is a unit distance from (x,y) and some of the neighbors of P lie outside the digital image of (x,y) is on the border if the image . The four diagonal neighbor of P have coordinated
(x+1,y+1),(x+1,y+1),(x-1,y+1),(x-1,y-1)

And are deported by nd (p) .these points, together with the 4-neighbours are called 8 – neighbors of P denoted by ns(p)

1.1.13.2 Adjacency

Let v be the set of gray –level values used to define adjacency ,in a binary image ,v={1} if we are reference to adjacency of pixel with value. Three types of adjacency

4- Adjacency – two pixel P and Q with value from V are 4 –adjacency if A is in the set n4(P)

8- Adjacency – two pixel P and Q with value from V are 8 –adjacency if A is in the set n8(P)

M-adjacency –two pixel P and Q with value from V are m – adjacency if

(i) Q is in n4 (p) or
(ii)Q is in nd (q) and the set N4(p) U N4(q) has no pixel whose values are from V

1.1.13.3 Distance measures

For pixel p,q and z with coordinate (x.y) ,(s,t) and (v,w) respectably D is a distance function or metric if
D [p.q] ≥ O {D[p.q] = O iff p=q}
D [p.q] = D [p.q] and
D [p.q] ≥ O {D[p.q]+D(q,z)

The Education Distance between p and is defined
as De (p,q) = Iy – t I

The D4 Education Distance between p and is defined
as De (p,q) = Iy – t I

UNIT-2 IMAGEENHANCEMENT IN FREQUENCY&SPATIAL DOMAIN

1.2 IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

1.2.1 Fourier Transform and the Frequency Domain

Any function that periodically reports itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient, this sum is called Fourier series. Even the functions which are non periodic but whose area under the curve if finite can also be represented in such form; this is now called Fourier transform.

A function represented in either of these forms and can be completely reconstructed via an inverse process with no loss of information.

1.2.1.1 1-D Fourier Transformation and its Inverse

If there is a single variable, continuous function f(x) , then Fourier transformation F (u) may be given as

$$\mathcal{F}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x)\exp(-j2\pi ux)dx \qquad j=\sqrt{-1}$$

And the reverse process to recover f(x) from F(u) is

$$\mathcal{F}^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u)\exp[j2\pi ux]du$$

Equation (a) and (b) comprise of Fourier transformation pair.

Fourier transformation of a discrete function of one variable f(x), x=0, 1, 2, m-1 is given by

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\exp[-j2\pi ux/N] \quad \text{for } u=0,1,2,\ldots,N-1$$

to obtain f(x) from F(u)

$$f(x) = \sum_{u=0}^{N-1} F(u)\exp[j2\pi ux/N] \quad \text{for } x=0,1,2,\ldots,N-1$$

The above two equation (e) and (f) comprise of a discrete Fourier transformation pair. According to Euler's formula

$$e^{jx} = \cos x + j\sin x$$

Substituting these value to equation (e)

F(u)=**Σ**f(x)[cos 2**π**ux/N+jSin 2**π**ux/N] for u=0,1,2,……,N-1

Now each of the m terms of F(u) is called a frequency component of transformation

"The Fourier transformation separates a function into various components, based on frequency components. These components are complex quantities.

F(u) in polar coordinates

$$F(u) = R(u) + jI(u) \qquad \text{or} \qquad F(u) = |F(U)|e^{j\phi(u)}$$

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2} \qquad \phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right]$$

1.2.1.2 2-D Fourier Transformation and its Inverse

The Fourier Transform of a two dimensional continuous function f(x,y) (an image) of size M * N is given by

$$\mathcal{F}\{f(x,y)\} = F(u,v) = \iint_{-\infty} f(x,y)\exp[-j2\pi(ux+vy)]dxdy$$

Inverse Fourier transformation is given by equation

$$\mathcal{F}^{-1}\{F(u,v)\} = f(x,y) = \iint_{-\infty}^{\infty} F(u,v)\exp[j2\pi(ux+vy)]dudv$$

Where (u,v) are frequency variables.
Preprocessing is done to shift the origin of F(u,v) to frequency coordinate (m/2,n/2) which is the center of the M*N area occupied by the 2D-FT. It is known as frequency rectangle.

It extends form u =0 to M-1 and v=0 to N-1. For this, we multiply the input image by $(-1)^{x+y}$ prior to compute the transformation

$$\mathcal{F}\{f(x,y)\,(-1)^{x+y}\} = F(u-M/2, v-N/2)$$

$\mathcal{F}(.)$ denotes the Fourier transformation of the argument Value of transformation at (u,v)=(0,0) is

$$F(0,0)=1/MN\Sigma\Sigma f(x,y)$$

1.2.1.3 Discrete Fourier Transform

$$\{f(x_0), f(x_0+\triangle x), \ldots \ldots f(x_0+[N-1]\triangle x)$$

$$\Longrightarrow f(x)=f(x_0+x\,\triangle x)$$

f(0),f(1),f(2)……f(N-1) denotes any N uniformly spaced samples.

$$\text{DFT}\ \ F(u)=\frac{1}{N}\sum_{x=0}^{N-1} f(x)\exp[-j2\pi ux/N]\ \ \text{for } u=0,1,2,\ldots..,N-1$$

Extending it to two variables

$$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp\left(-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$

for u=0,1,2,….,M-1 v=0,1,2,…N-1

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{u=0}^{N-1} F(u,v)\exp\left(j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$

for x=0,1,…,M-1 y=0,1,…..,N-1

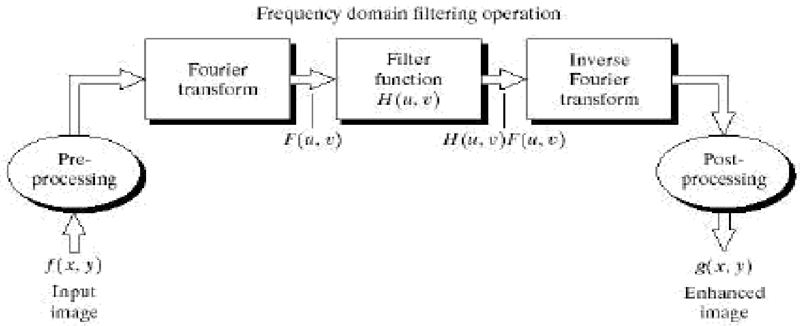
$$\Delta u=\frac{1}{M\Delta x} \qquad \Delta v=\frac{1}{N\Delta y}$$

UNIT -2

1.2.2 **<u>Basis of Filtering in Frequency Domain</u>**

Basic steps of filtering in frequency Domain

i) Multiply the input image by $(-1)^{X+Y}$ to centre the transform
ii) Compute F(u,v), Fourier Transform of the image
iii) Multiply f(u,v) by a filter function H(u,v)
iv) Compute the inverse DFT of Result of (iii)
v) Obtain the real part of result of (iv)
vi) Multiply the result in (v) by $(-1)^{x=y}$



H(u,v) called a filter because it suppresses certain frequencies from the image while leaving others unchanged.

1.2.3 **Filters**

1.2.3.1 Smoothing Frequency Domain Filters

Edges and other sharp transition of the gray levels of an image contribute significantly to the high frequency contents of its Fourier transformation. Hence smoothing is achieved in the frequency domain by attenuation a specified range of high frequency components in the transform of a given image.
Basic model of filtering in the frequency domain is

$$G(u,v) = H(u,v)F(u,v)$$

F(u,v) - Fourier transform of the image to be smoothed

Objective is to find out a filter function H (u,v) that yields G (u,v) by attenuating the high frequency component of F (u,v)
There are three types of low pass filters

1. Ideal
2. Butterworth
3. Gaussian

1.2.3.1.1 IDEAL LOW PASS FILTER

It is the simplest of all the three filters
It cuts of all high frequency component of the Fourier transform that are at a distance greater that a specified distance D_0 form the origin of the transform.
it is called a two – dimensional ideal low pass filter (ILPF) and has the transfer function

$$H(u,v) = \begin{cases} 1 \text{ if } D(u,v) \le D_0 \\ 0 \text{ if } D(u,v) > D_0 \end{cases}$$

Where D_0 is a specified nonnegative quantity and D(u,v) is the distance from point (u,v) to the center of frequency rectangle
If the size of image is M*N , filter will also be of the same size so center of the frequency rectangle (u,v) = (M/2, N/2) because of center transform

$$D(u,v) = (u^2 + v^2)^{1/2}$$

Because it is ideal case. So all frequency inside the circle are passed without any attenuation where as all frequency outside the circle are completely attenuated

For an ideal low pass filter cross section, the point of transition between H (u,v) =1 and H (u,v)=0 is called of the “ cut of frequency

One way to establish a set of standard cut of frequency locus is to compute circle that include specified amount of total image Power Pt

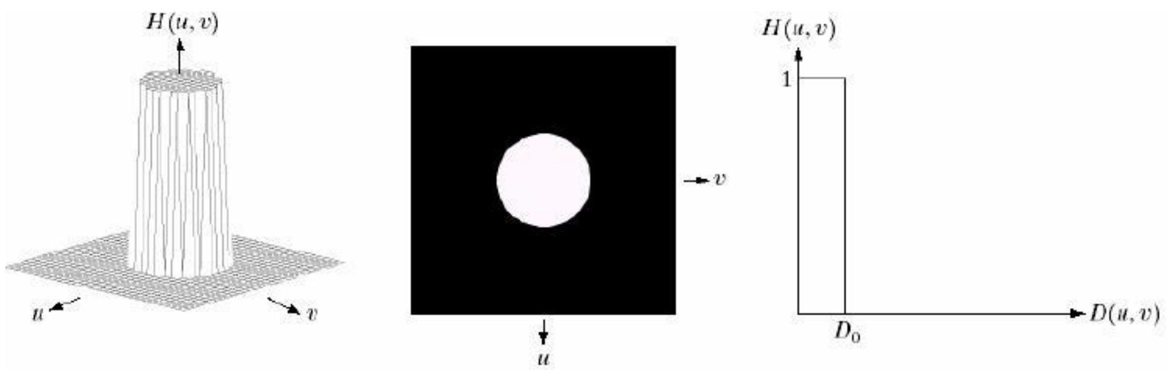
$$|100[\sum_{u}\sum_{v} P(u,v)/P_T$$

It can be obtained by summing the components of the power spectrum at each point (u,v) for u=0,1,2,3,4,..,,,,,,,,,,,,,,,,, N-1.

If transform has been centered a circle of radius r with origin at the center of the frequency rectangle encloses ∞ percent of the power

For R = 5 ∞ = 92 % most blurred image because all sharp details are removed
R = 15 ∞ = 94.6 %
R = 30 ∞ = 96.4 %
R = 80 ∞ = 98 % maximum ringing only 2 % power is removed
R = 230 ∞ = 99.5 % very slight blurring only 0.5 % power is removed

ILPF is not suitable for practical usage. But they can be implemented in any computer system



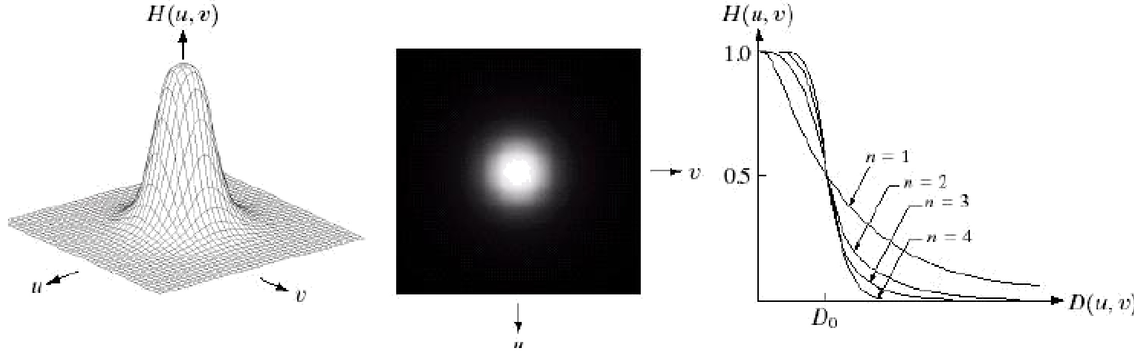
1.2.3.1.2 <u>BUTTERWORTH LOW PASS FILTER</u>

It has a parameter called the filter order.
For high values of filter order it approaches the form of the ideal filter whereas for low filter order values it reach Gaussian filter. It may be viewed as a transition between two extremes.
The transfer function of a Butterworth low pass filter (BLPF) of order n with cut off frequency at distance Do from the origin is defined as

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$

Most appropriate value of n is 2.
It does not have sharp discontinuity unlike ILPF that establishes a clear cutoff between passed and filtered frequencies.
Defining a cutoff frequency is a main concern in these filters. This filter gives a smooth transition in blurring as a function of increasing cutoff frequency. A Butterworth filter of order 1 has no ringing.

Ringing increases as a function of filter order. (Higher order leads to negative values)



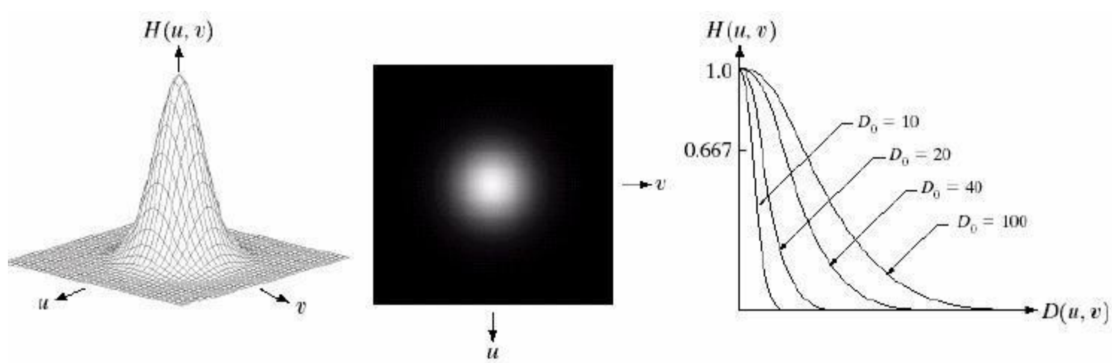
1.2.3.1.3 GAUSSIAN LOW PASS FILTER

The transfer function of a Gaussian low pass filter is

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}$$

Where:
D(u,v)- the distance of point (u,v) from the center of the transform
σ = D_0- specified cut off frequency

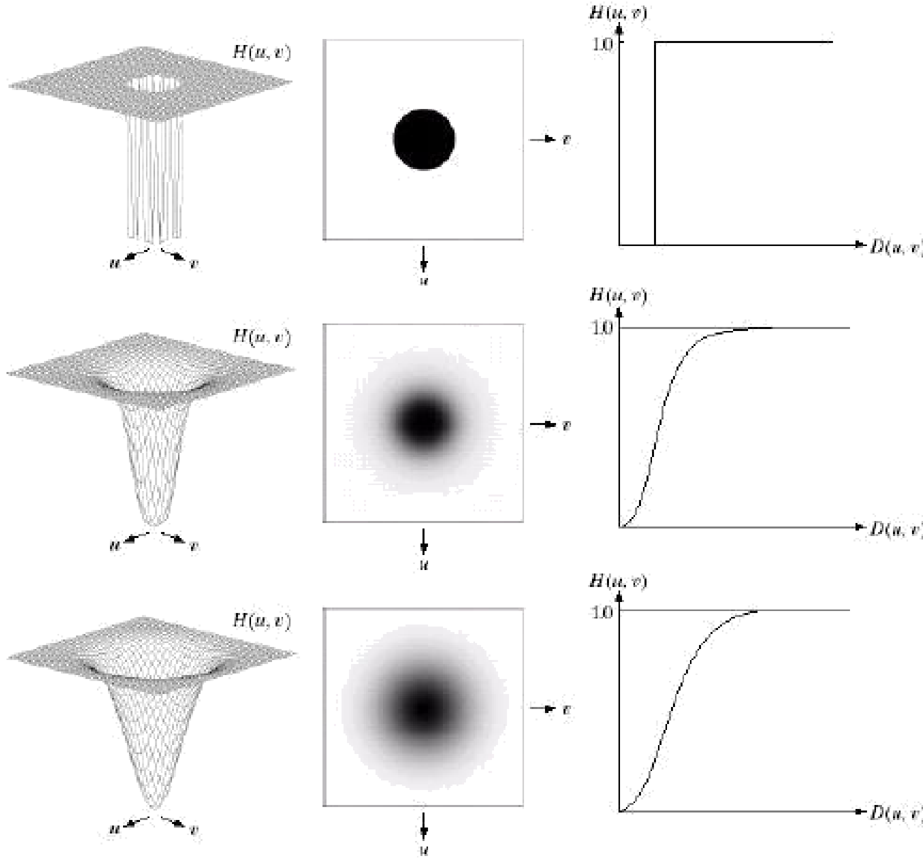
The filter has an important characteristic that the inverse of it is also Gaussain.



1.2.3.2 SHARPENING FREQUENCY DOMAIN FILTERS

Image sharpening can be achieved by a high pass filtering process, which attenuates the low-frequency components without disturbing high-frequency information. These are radially symmetric and completely specified by a cross section.
If we have the transfer function of a low pass filter the corresponding high pass filter can be obtained using the equation

$$H_{hp}(u,v)=1- H_{lp}(u,v)$$



1.2.3.2.1 IDEAL HIGH PASS FILTER

This filter is opposite of the Ideal Low Pass filter and has the transfer function of the form

$$H(u,v)=\begin{cases}0 & \text{if } D(u,v)\le D_0\\ 1 & \text{if } D(u,v)>D_0\end{cases}$$

1.2.3.2.2 BUTTERWORTH HIGH PASS FILTER

The transfer function of Butterworth High Pass filter of order n is given by the equation

$$H(u,v)=\frac{1}{1+[D_0/D(u,v)]^{2n}}$$

1.2.3.2.3 GAUSSIAN HIGH PASS FILTER

The transfer function of a Gaussain High Pass Filter is given by the equation

$$H(u,v)=1-e^{-D^2(u,v)/2\sigma^2}$$

1.2.4 **Homomorphic Filtering**

Homomorphic filters are widely used in image processing for compensating the effect of no uniform illumination in an image. Pixel intensities in an image represent the light reflected from the corresponding points in the objects. As per as image model, image f(z,y) may be characterized by two components: (1) the amount of source light incident on the scene being viewed, and (2) the amount of light reflected by the objects in the scene. These portions of light are called the illumination and reflectance components, and are denoted i (x , y) and r (x , y) respectively. The functions i (x , y) and r (x , y) combine multiplicatively to give the image function f (x , y):

$$f(x, y) = i(x, y).r(x, y) \qquad (1)$$

where $0 < i(x, y) < a$ and $0 < r(x, y) < 1$. Homomorphic filters are used in such situations where the image is subjected to the multiplicative interference or noise as depicted in Eq. 1. We cannot easily use the above product to operate separately on the frequency components of illumination and reflection because the Fourier transform of f (x , y) is not separable; that is

$$F[f(x,y)) \text{ not equal to } F[i(x, y)].F[r(x, y)].$$

We can separate the two components by taking the logarithm of the two sides

$$\ln f(x,y) = \ln i(x, y) + \ln r(x, y).$$

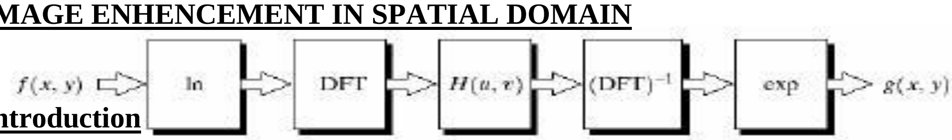
Taking Fourier transforms on both sides we get,

$$F[\ln f(x,y)\} = F[\ln i(x, y)\} + F[\ln r(x, y)].$$

that is, F(x,y) = I(x,y) + R(x,y), where F, I and R are the Fourier transforms ln f(x,y),ln i(x, y) , and ln r(x, y). respectively. The function F represents the Fourier transform of the sum of two images: a low-frequency illumination image and a high-frequency reflectance image. If we now apply a filter with a transfer function that suppresses low- frequency components and enhances high-frequency components, then we can suppress the illumination component and enhance the reflectance component. Taking the inverse transform of F (x , y) and then anti-logarithm, we get

$$f'(x, y) = i'(x, y) + r'(x, y)$$

2.1 **IMAGE ENHENCEMENT IN SPATIAL DOMAIN**



2.1.1 **Introduction**

The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application. Image enhancement approaches fall into two board categories

- ⇨ Spatial domain methods
- ⇨ Frequency domain methods

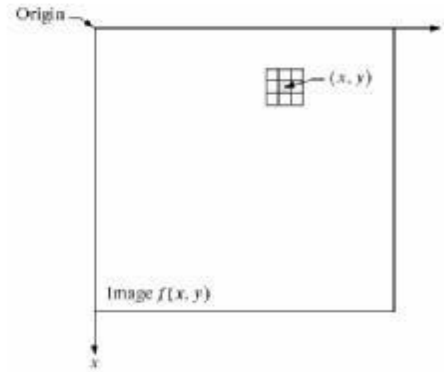
The term spatial domain refers to the image plane itself and approaches in this categories are based on direct manipulation of pixel in an image. Spatial domain process are denoted by the expression

g(x,y)=T[f(x,y)]

f(x,y)- input image g(x,y)-processed image

T- operator on f, defined over some neighborhood of f(x,y)

The neighborhood of a point (x,y) can be explain by using as square or rectangular sub image area centered at (x,y).



The center of sub image is moved from pixel to pixel starting at the top left corner. The operator T is applied to each location (x,y) to find the output g at that location . The process utilizes only the pixel in the area of the image spanned by the neighborhood.

2.1.2 **Basic Gray Level Transformation Functions**

It is the simplest form of the transformations when the neighborhood is of size IXI. In this case g depends only on the value of f at (x,y) and T becomes a gray level transformation function of the forms

$$S=T(r)$$

r- Denotes the gray level of f(x,y)

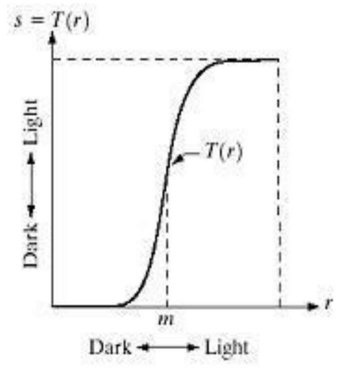
s- Denotes the gray level of g(x,y) at any point (x,y)

Because enhancement at any point in an image deepens only on the gray level at that point, technique in this category are referred to as point processing.
There are basically three kinds of functions in gray level transformation –

2.1.2.1 Point Processing

2.1.2.1.1 Contract stretching -

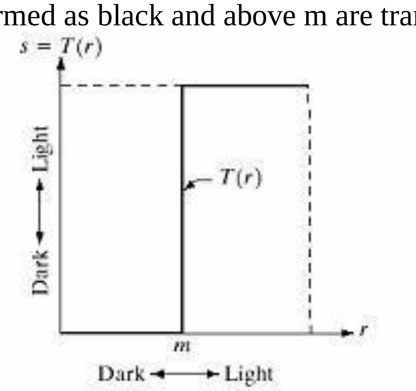
It produces an image of higher contrast than the original one.
The operation is performed by darkening the levels below m and brightening the levels above m in the original image.



In this technique the value of r below m are compressed by the transformation function into a narrow range of s towards black .The opposite effect takes place for the values of r above m.

2.1.2.1.2 Thresholding function -

It is a limiting case where T(r) produces a two levels binary image.
The values below m are transformed as black and above m are transformed as white.



2.1.2.2 Basic Gray Level Transformation

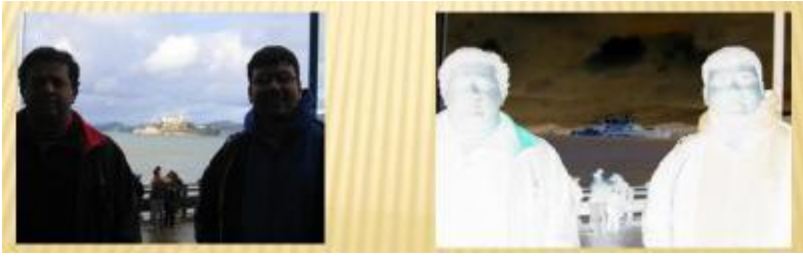
These are the simplest image enhancement techniques.

2.1.2.2.1 Image Negative –

The negative of in image with gray level in the range [0, l-1] is obtained by using the negative transformation.
The expression of the transformation is

$$s = L-1-r$$

Reverting the intensity levels of an image in this manner produces the equivalent of a photographic negative. This type of processing is practically suited for enhancing white or gray details embedded in dark regions of an image especially when the black areas are dominant in size.



2.1.2.2.2 Log transformations

The general form of the log transformation is

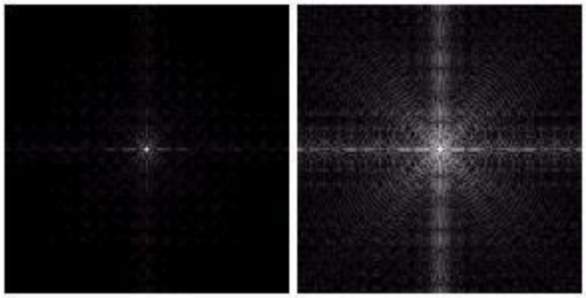
$$s = c \log(1+r)$$

Where c- constant

R $\geq$ 0

This transformation maps a narrow range of gray level values in the input image into a wider range of output gray levels. The opposite is true for higher values of input levels. We would use this transformations to expand the values of dark pixels in an image while compressing the higher level values. The opposite is true for inverse log transformation.

The log transformation function has an important characteristic that it compresses the dynamic range of images with large variations in pixel values. Eg- Fourier spectrum



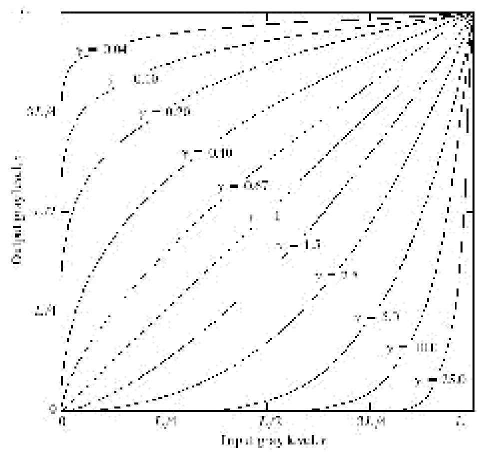
2.1.2.2.3 Power Law Transformation

Power law transformations has the basic form

$$S = cr^{y}$$

Where c and y are positive constants.

Power law curves with fractional values of y map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input gray levels. We may get various curves by varying values of y.



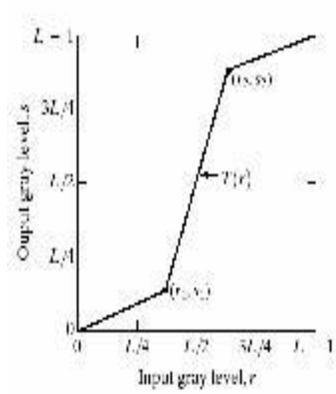
A variety of devices used for image capture, printing and display respond according to a power law. The process used to correct this power law response phenomenon is called gamma correction.
For eg-CRT devices have intensity to voltage response that is a power function.
Gamma correction is important if displaying an image accurately on a computer screen is of concern. Images that are not corrected properly can look either bleached out or too dark.
Color phenomenon also uses this concept of gamma correction. It is becoming more popular due to use of images over the internet.
It is important in general purpose contract manipulation. To make an image black we use y>1 and y<1 for white image.

2.1.2.3 Piece wise Linear transformation functions

The principal advantage of piecewise linear functions is that these functions can be arbitrarily complex. But their specification requires considerably more user input.

2.1.2.3.1 Contrast Stretching

It is the simplest piecewise linear transformation function.
We may have various low contrast images and that might result due to various reasons such as lack of illumination, problem in imaging sensor or wrong setting of lens aperture during image acquisition.
The idea behind contrast stretching is to increase the dynamic range of gray levels in the image being processed.



The location of points (r1,s1) and (r_2,s_2) control the shape of the curve

a) If r_1=r_2 and s_1=s_2, the transformation is a linear function that deduces no change in gray levels.

b) If r1=s1, s1=0 , and s2=L-1, then the transformation become a thresholding function that creates a binary image

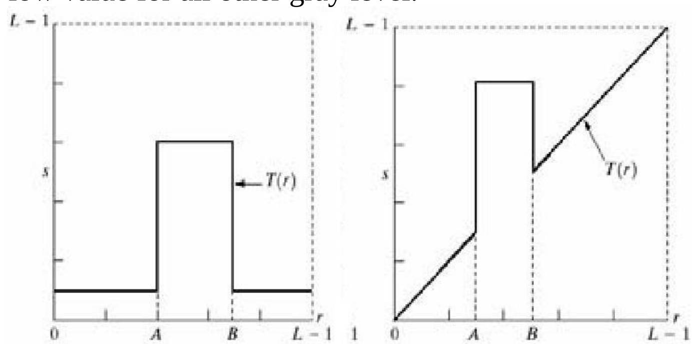
c) Intermediate values of (r1, s1) and (r2, s2) produce various degrees of spread in the gray value of the output image thus effecting its contract.

Generally r1≤ r2 and s1 ≤ s2 so that the function is single valued and monotonically increasing

2.1.2.3.2 Gray Level Slicing

Highlighting a specific range of gray levels in an image is often desirable
For example when enhancing features such as masses of water in satellite image and enhancing flaws in x- ray images.
There are two ways of doing this-

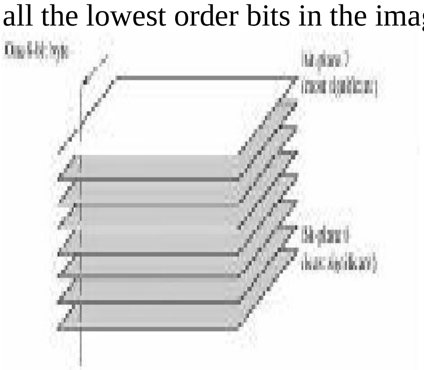
(1) One method is to display a high value for all gray level in the range. Of interest and a low value for all other gray level.

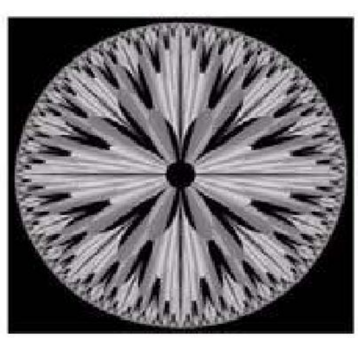


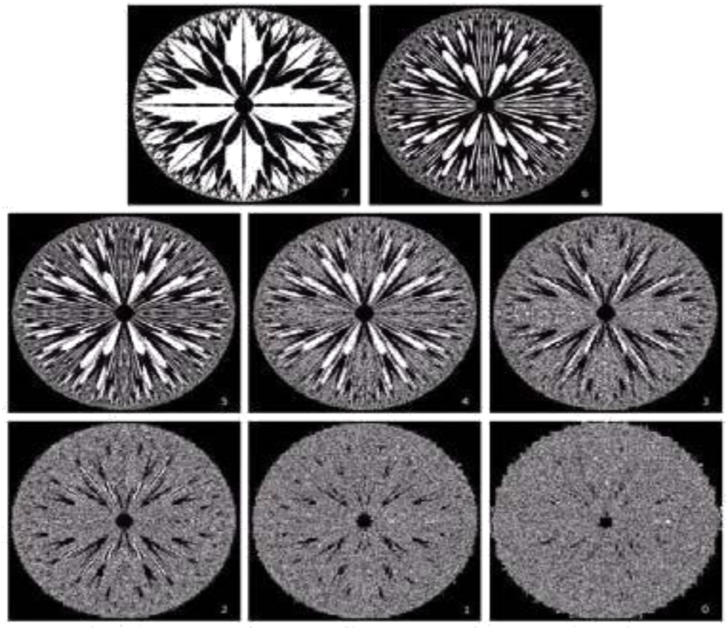
(2) Second method is to brighten the desired ranges of gray levels but preserve the background and gray level tonalities in the image.

2.1.2.3.3 Bit Plane Slicing

Sometimes it is important to highlight the contribution made to the total image appearance by specific bits. Suppose that each pixel is represented by 8 bits.
Imagine that an image is composed of eight 1-bit planes ranging from bit plane 0 for the least significant bit to bit plane 7 for the most significant bit. In terms of 8-bit bytes, plane 0 contains all the lowest order bits in the image and plane 7 contains all the high order bits.







.
High order bits contain the majority of visually significant data and contribute to more subtle details in the image.

Separating a digital image into its bits planes is useful for analyzing the relative importance played by each bit of the image.

It helps in determining the adequacy of the number of bits used to quantize each pixel. It is also useful for image compression.

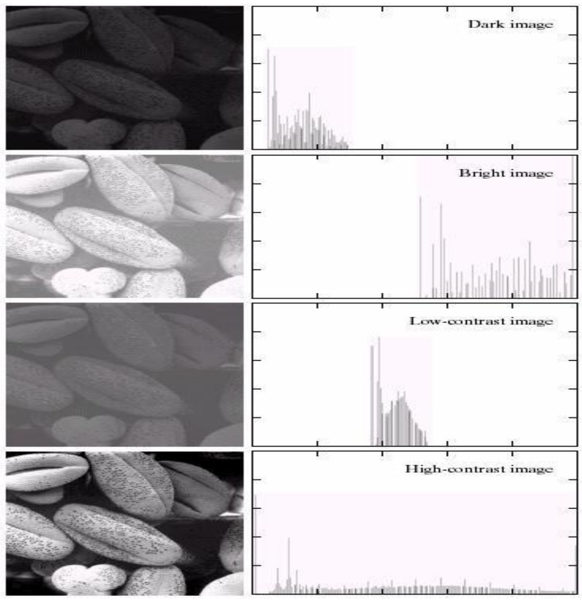
2.1.3 **Histogram Processing**

The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function of the form

$$H(rk)=nk$$

where rk is the kth gray level and nk is the number of pixels in the image having the level rk.. A normalized histogram is given by the equation

$p(rk)=nk/n$ for $k=0,1,2,\ldots,L-1$ P(rk) gives the estimate of the probability of occurrence of gray level rk. The sum of all components of a normalized histogram is equal to 1. The histogram plots are simple plots of H(rk)=nk versus rk.



In the dark image the components of the histogram are concentrated on the low (dark) side of the gray scale. In case of bright image the histogram components are baised towards the high side of the gray scale.
The histogram of a low contrast image will be narrow and will be centered towards the middle of the gray scale.
The components of the histogram in the high contrast image cover a broad range of the gray scale. The net effect of this will be an image that shows a great deal of gray levels details and has high dynamic range.

2.1.3.1 Histogram Equalization

Histogram equalization is a common technique for enhancing the appearance of images. Suppose we have an image which is predominantly dark. Then its histogram would be skewed towards the lower end of the grey scale and all the image detail are compressed into the dark end of the histogram. If we could 'stretch out' the grey levels at the dark end to produce a more uniformly distributed histogram then the image would become much clearer.
Let there be a continuous function with r being gray levels of the image to be enhanced.
The range of r is [0, 1] with r=0 repressing black and r=1 representing white.
The transformation function is of the form

$$S=T(r) \qquad \text{where } 0<r<1$$

It produces a level s for every pixel value r in the original image.
The transformation function is assumed to fulfill two condition
T(r) is single valued and monotonically increasing in the internal

$$0<T(r)<1 \text{ for } 0<r,1$$

The transformation function should be single valued so that the inverse transformations should exist. Monotonically increasing condition preserves the increasing order from black to white in the output image.The second conditions guarantee that the output gray levels will be in the same range as the input levels.
The gray levels of the image may be viewed as random variables in the interval [0.1]
The most fundamental descriptor of a random variable is its probability density function (PDF)
Pr(r) and Ps(s) denote the probability density functions of random variables r and s respectively.
Basic results from an elementary probability theory states that if Pr(r) and Tr are known and T-1 (s) satisfies conditions (a), then the probability density function Ps(s) of the transformed variable s is given by the formula-

$$P_s(s) = P_r(r)\frac{dr}{ds},$$

Thus the PDF of the transformed variable s is the determined by the gray levels PDF of the input image and by the chosen transformations function.
A transformation function of a particular importance in image processing

$$s = T(r) = \int_0^r P_r(w)dw.$$

This is the cumulative distribution function of r.

Using this definition of T we see that the derivative of s with respect to r is

$$\frac{ds}{dr} = P_r(r).$$

Substituting it back in the expression for Ps we may get

$$P_s(s) = P_r(r)\frac{1}{P_r(r)} = 1$$

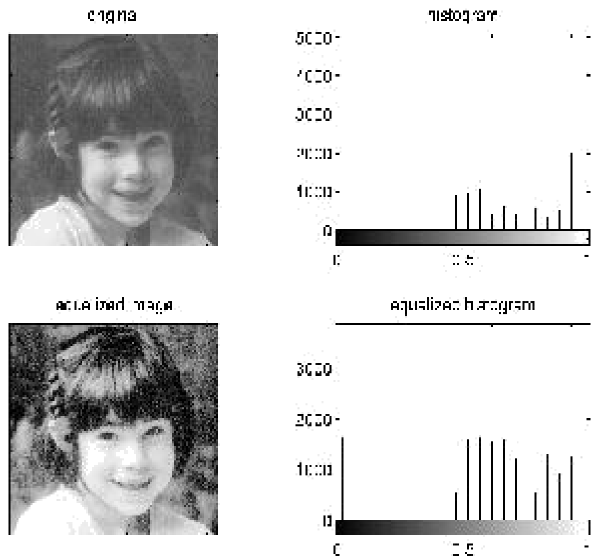
An important point here is that Tr depends on Pr(r) but the resulting Ps(s) always is uniform, and independent of the form of P(r).
For discrete values we deal with probability and summations instead of probability density functions and integrals.
The probability of occurrence of gray levels rk in an image as approximated

$$Pr(r) = nk/N$$

N is the total number of the pixels in an image.
nk is the number of the pixels that have gray level rk.
L is the total number of possible gray levels in the image.
The discrete transformation function is given by

$$s_k = T(r_k) = \sum_{i=0}^{k} \frac{n_i}{N} = \sum_{i=0}^{k} P_r(r_i).$$

Thus a processed image is obtained by mapping each pixel with levels rk in the input image into a corresponding pixel with level sk in the output image.
A plot of Pr (rk) versus rk is called a histogram. The transformation function given by the above equation is the called histogram equalization or linearization.
Given an image the process of histogram equalization consists simple of implementing the transformation function which is based information that can be extracted directly from the given image, without the need for further parameter specification.



Equalization automatically determines a transformation function that seeks to produce an output image that has a uniform histogram. It is a good approach when automatic enhancement is needed

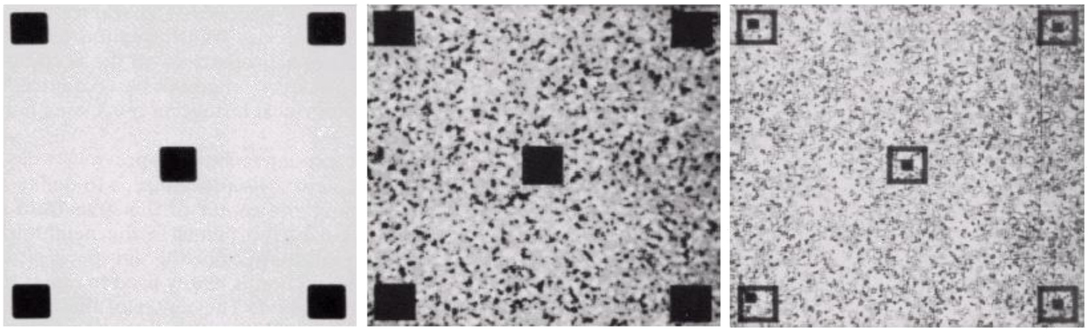
2.1.3.2 Histogram Matching (Specification)

In some cases it may be desirable to specify the shape of the histogram that we wish the processed image to have.
Histogram equalization does not allow interactive image enhancement and generates only one result: an approximation to a uniform histogram. Sometimes we need to be able to specify particular histogram shapes capable of highlighting certain gray-level ranges. The method use to generate a processed image that has a specified histogram is called histogram matching or histogram specification.
Algorithm
1. Compute sk=Pf (k), k = 0, …, L-1, the cumulative normalized histogram of f .
2. Compute G(k), k = 0, …, L-1, the transformation function, from the given histogram hz 3. Compute G-1(sk) for each k = 0, …, L-1 using an iterative method (iterate on z), or in effect, directly compute G-1(Pf (k))
4. Transform f using G-1(Pf (k)) .

2.1.4 **Local Enhancement**

In earlier methods pixels were modified by a transformation function based on the gray level of an entire image. It is not suitable when enhancement is to be done in some small areas of the image. This problem can be solved by local enhancement where a transformation function is applied only in the neighborhood of pixels in the interested region.
Define square or rectangular neighborhood (mask) and move the center from pixel to pixel. For each neighborhood

1) Calculate histogram of the points in the neighborhood
2) Obtain histogram equalization/specification function
3) Map gray level of pixel centered in neighborhood
4) The center of the neighborhood region is then moved to an adjacent pixel location and the procedure is repeated.

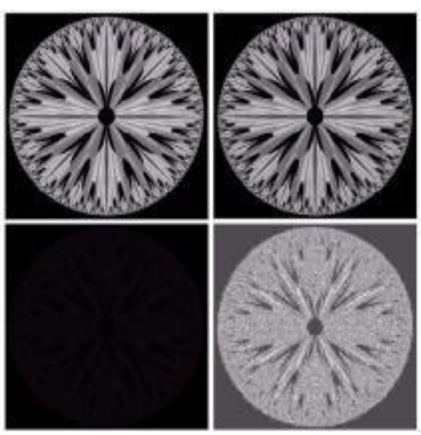


2.1.5 **Enhancement Using Arithmetic/Logic Operations**

These operations are performed on a pixel by basis between two or more images excluding not operation which is performed on a single image. It depends on the hardware and/or software that the actual mechanism of implementation should be sequential, parallel or simultaneous. Logic operations are also generally operated on a pixel by pixel basis.
Only AND, OR and NOT logical operators are functionally complete. Because all other operators can be implemented by using these operators.
While applying the operations on gray scale images, pixel values are processed as strings of binary numbers.
The NOT logic operation performs the same function as the negative transformation.
Image Masking is also referred to as region of Interest (Ro1) processing. This is done to highlight a particular area and to differentiate it from the rest of the image.
Out of the four arithmetic operations, subtraction and addition are the most useful for image enhancement.

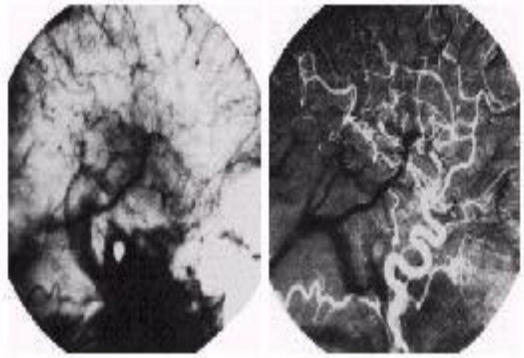
2.1.5.1 Image Subtraction

The difference between two images f(x,y) and h(x,y) is expressed
as g(x,y)= f(x,y)-h(x,y)
It is obtained by computing the difference between all pairs of corresponding pixels from f and h.
The key usefulness of subtraction is the enhancement of difference between images.
This concept is used in another gray scale transformation for enhancement known as bit plane slicing The higher order bit planes of an image carry a significant amount of visually relevant detail while the lower planes contribute to fine details.
It we subtract the four least significant bit planes from the image The result will be nearly identical but there will be a slight drop in the overall contrast due to less variability in the gray level values of image .



The use of image subtraction is seen in medical imaging area named as mask mode radiography. The mask h (x,y) is an X-ray image of a region of a patient's body this image is captured by using as intensified TV camera located opposite to the x-ray machine then a consistent medium is injected into the patient's blood storm and then a series of image are taken of the region same as h(x,y).

The mask is then subtracted from the series of incoming image. This subtraction will give the area which will be the difference between f(x,y) and h(x,y) this difference will be given as enhanced detail in the output image.



This procure produces a move shoving now the contrast medium propagates through various arteries of the area being viewed.

Most of the image in use today is 8- bit image so the values of the image lie in the range 0 to 255.

The value in the difference image can lie from -255 to 255. For these reasons we have to do some sort of scaling to display the results

There are two methods to scale an image

(i) Add 255 to every pixel and then divide at by 2.

This gives the surety that pixel values will be in the range 0 to 255 but it is not guaranteed whether it will cover the entire 8 – bit range or not.

It is a simple method and fast to implement but will not utilize the entire gray scale range to display the results.

(ii) Another approach is

(a) Obtain the value of minimum difference

(b) Add the negative of minimum value to the pixels in the difference image(this will give a modified image whose minimum value will be 0)

(c) Perform scaling on the difference image by multiplying each pixel by the quantity 255/max. This approach is complicated and difficult to implement.

Image subtraction is used in segmentation application also

2.1.5.2 <u>Image Averaging</u>

Consider a noisy image g(x,y) formed by the addition of noise n(x,y) to the original image f(x,y)

$$g(x,y) = f(x,y) + n(x,y)$$

Assuming that at every point of coordinate (x,y) the noise is uncorrelated and has zero average value

The objective of image averaging is to reduce the noise content by adding a set of noise images, {gi(x,y)}

If in image formed by image averaging K different noisy images

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^{K} g_i(x, y)$$

$$E\{\bar{g}(x, y)\} = f(x,y)$$

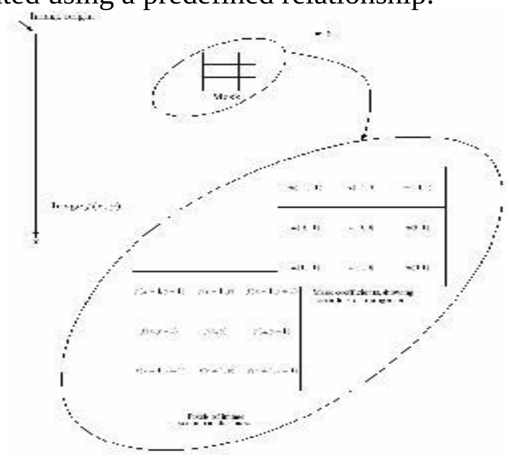
As k increases the variability (noise) of the pixel value at each location (x,y) decreases
E{g(x,y)} = f(x,y) means that g(x,y) approaches f(x,y) as the number of noisy image used in the averaging processes increases

Image averaging is important in various applications such as in the field of astronomy where the images are low light levels

2.1.6 Basic of Spatial Filtering

Spatial filtering is an example of neighborhood operations, in this the operations are done on the values of the image pixels in the neighborhood and the corresponding value of a sub image that has the same dimensions as of the neighborhood

This sub image is called a filter, mask, kernel, template or window; the values in the filter sub image are referred to as coefficients rather than pixel. Spatial filtering operations are performed directly on the pixel values (amplitude/gray scale) of the image
The process consists of moving the filter mask from point to point in the image. At each point (x,y) the response is calculated using a predefined relationship.



For linaer spatial filtering the response is given by a sum of products of the filter coefficient and the corresponding image pixels in the area spanned by the filter mask.
The results R of liner filtering with the filter mask at point (x,y) in the image is

$$R = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \cdots + w(0,0)f(x,y) + \cdots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)$$

The sum of products of the mask coefficient with the corresponding pixel directly under the mask. The coefficient w (0,0) coincides with image value f(x,y) indicating that mask it centered at (x,y) when the computation of sum of products takes place

For a mask of size MxN we assume m=2a+1 and n=2b+1, where a and b are nonnegative integers.
It shows that all the masks are of add size.

In the general liner filtering of an image of size f of size M*N with a filter mask of size m*m is given by the expression

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$

Where a= (m-1)/2 and b = (n-1)/2

To generate a complete filtered image this equation must be applied for x=0, 1, 2, -----M-1 and y=0,1,2---,N-1. Thus the mask processes all the pixels in the image.

The process of linear filtering is similar to frequency domain concept called convolution. For this reason, linear spatial filtering often is referred to as convolving a mask with an image. Filter mask are sometimes called convolution mask.

$$R= W_1Z_1+W_2 Z_2 +\ldots+ W_{mn} Z_{mn}$$

Where w's are mask coefficients and

z's are the values of the image gray levels corresponding to those coefficients.

mn is the total number of coefficients in the mask.

An important point in implementing neighborhood operations for spatial filtering is the issue of what happens when the center of the filter approaches the border of the image. There are several ways to handle this situation.

i) To limit the excursion of the center of the mask to be at distance of less than (n-1) /2 pixels form the border. The resulting filtered image will be smaller than the original but all the pixels will be processed with the full mask.

ii) Filter all pixels only with the section of the mask that is fully contained in the image. It will create bands of pixels near the border that will be processed with a partial mask.

iii) Padding the image by adding rows and columns of o's & or padding by replicating rows and columns. The padding is removed at the end of the process.

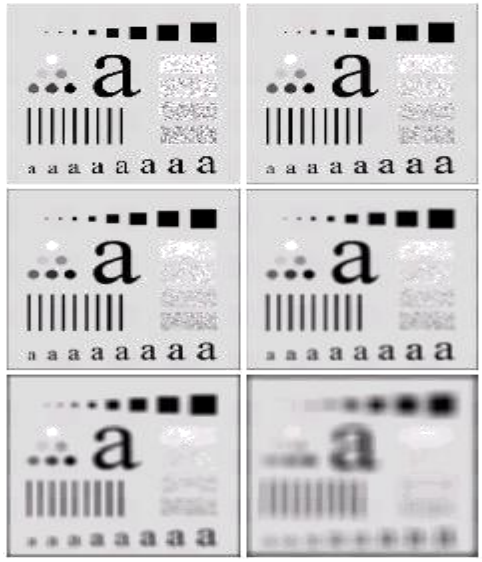
2.1.6.1 Smoothing Spatial Filters

These filters are used for blurring and noise reduction blurring is used in preprocessing steps such as removal of small details from an image prior to object extraction and bridging of small gaps in lines or curves.

2.1.6.1.1 Smoothing Linear Filters

The output of a smoothing liner spatial filter is simply the average of the pixel contained in the neighborhood of the filter mask. These filters are also called averaging filters or low pass filters.

The operation is performed by replacing the value of every pixel in the image by the average of the gray levels in the neighborhood defined by the filter mask. This process reduces sharp transitions in gray levels in the image.



A major application of smoothing is noise reduction but because edge are also provided using sharp transitions so smoothing filters have the undesirable side effect that they blur edges . It also removes an effect named as false contouring which is caused by using insufficient number of gray levels in the image.
Irrelevant details can also be removed by these kinds of filters, irrelevant means which are not of our interest.

A spatial averaging filter in which all coefficients are equal is sometimes referred to as a "box filter"

$$\frac{1}{9}\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}\qquad\frac{1}{16}\times\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

A weighted average filter is the one in which pixel are multiplied by different coefficients.

2.1.6.1.2 <u>Order Statistics Filter</u>

These are nonlinear spatial filter whose response is based on ordering of the pixels contained in the image area compressed by the filter and the replacing the value of the center pixel with value determined by the ranking result.

The best example of this category is median filter. In this filter the values of the center pixel is replaced by median of gray levels in the neighborhood of that pixel. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters.

These filters are particularly effective in the case of impulse or salt and pepper noise. It is called so because of its appearance as white and black dots superimposed on an image.

The median £ of a set of values is such that half the values in the set less than or equal to £ and half are greater than or equal to this. In order to perform median filtering at a point in an image, we first sort the values of the pixel in the question and its neighbors, determine their median and assign this value to that pixel.

We introduce some additional order-statistics filters. Order-statistics filters are spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter. The response of the filter at any point is determined by the ranking result

2.1.6.1.2.1 Median filter

The best-known order-statistics filter is the median filter, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel:

$$\hat{f}(x, y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s, t)\}.$$

The original value of the pixel is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise. In fact, the median filter yields excellent results for images corrupted by this type of noise.

2.1.6.1.2.2 Max and min filters

Although the median filter is by far the order-statistics filter most used in image processing.it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but the reader will recall from basic statis¬tics that ranking lends itself to many other possibilities. For example, using the 100th perccntile results in the so-called max filter given by:

$$\hat{f}(x, y) = \max_{(s,t)\in S_{xy}}\{g(s, t)\}.$$

This filter is useful for finding the brightest points in an image. Also, because pepper noise has very low values, it is reduced by this filter as a result of the max selection process in the subimage area S. The 0th percentile filter is the Min filter.

2.1.6.2 Sharpening Spatial Filters

The principal objective of sharpening is to highlight fine details in an image or to enhance details that have been blurred either in error or as a natural effect of particular method for image acquisition.
The applications of image sharpening range from electronic printing and medical imaging to industrial inspection and autonomous guidance in military systems.
As smoothing can be achieved by integration, sharpening can be achieved by spatial differentiation. The strength of response of derivative operator is proportional to the degree of discontinuity of the image at that point at which the operator is applied. Thus image differentiation enhances edges and other discontinuities and deemphasizes the areas with slow varying grey levels.
It is a common practice to approximate the magnitude of the gradient by using absolute values instead of square and square roots.
A basic definition of a first order derivative of a one dimensional function *f(x)* is the difference.

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

Similarly we can define a second order derivative as the difference

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x$$

2.1.6.2.1 <u>The LAPLACIAN</u>

The second order derivative is calculated using Laplacian. It is simplest isotropic filter. Isotropic filters are the ones whose response is independent of the direction of the image to which the operator is applied.

The Laplacian for a two dimensional function f(x,y) is defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Partial second order directive in the x-direction

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

And similarly in the y-direction

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

The digital implementation of a two-dimensional Laplacian obtained by summing the two components

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y). \quad ($$

The equation can be represented using any one of the following masks

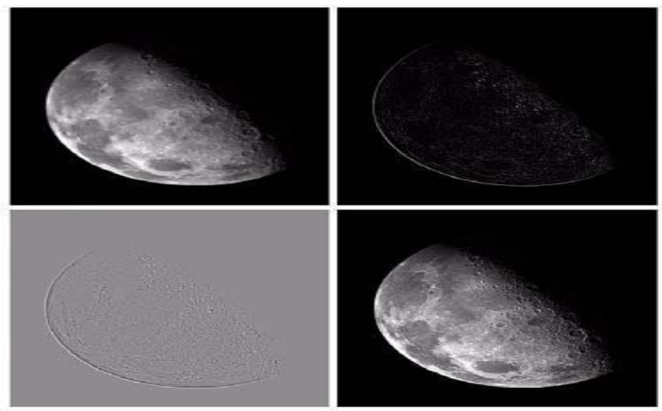
0	1	0	1	1	1
1	−4	1	1	−8	1
0	1	0	1	1	1
0	−1	0	−1	−1	−1
−1	4	−1	−1	8	−1
0	−1	0	−1	−1	−1

Laplacian highlights gray-level discontinuities in an image and deemphasize the regions of slow varying gray levels. This makes the background a black image. The background texture can be recovered by adding the original and Laplacian images.

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases}$$

$$\begin{aligned} g(x,y) &= f(x,y) - [f(x+1,y) + f(x-1,y) \\ &\quad + f(x,y+1) + f(x,y-1)] + 4f(x,y) \\ &= 5f(x,y) - [f(x+1,y) + f(x-1,y) \\ &\quad + f(x,y+1) + f(x,y-1)]. \end{aligned}$$

For example:



.

The strength of the response of a derivative operator is propositional to the degree of discontinuity of the image at that point at which the operator is applied. Thus image differentiation enhances eddies and other discontinuities and deemphasizes areas with slowly varying gray levels.

The derivative of a digital function is defined in terms of differences. Any first derivative definition

(1) Must be zero in flat segments (areas of constant gray level values)
(2) Must be nonzero at the onset of a gray level step or ramp
(3) Must be nonzero along ramps.

Any second derivative definition

(1) Must be zero in flat areas
(2) Must be nonzero at the onset and end of a gray level step or ramp
(3) Must be zero along ramps of constant slope .

It is common practice to approximate the magnitude of the gradient by using also lute values instead or squares and square roots:

Roberts Goss gradient operators

For digitally implementing the gradient operators
Let center point, 5z denote f(x,y), Z1 denotes f(x-1,y) and so on

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

But it different implement even sized mask. So the smallest filter mask is size 3x3 mask is

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

The difference between third and first row a 3x3 mask approximates the derivate in the x-direction and difference between the third and first column approximates the derivative in y-direction. These masks are called sobel operators.

2.1.7 Unsharp Masking and High Boost Filtering

Unsharp masking means subtracting a blurred version of an image form the image itself. Where f(x,y) denotes the sharpened image obtained by unsharp masking and f(x,y) is a blurred version of (x,y)

$$f_s(x,y) = f(x,y) - \overline{f}(x,y)$$

A slight further generalization of unsharp masking is called high boost filtering. A high boost filtered image is defined at any point (x,y) as

$$f_{hb}(x,y) = Af(x,y) - \overline{f}(x,y)$$

UNIT-3

IMAGE RESTORATION

3.1 IMAGE RESTORATION

Restoration improves image in some predefined sense. It is an objective process. Restoration attempts to reconstruct an image that has been degraded by using a priori knowledge of the degradation phenomenon. These techniques are oriented toward modeling the degradation and then applying the inverse process in order to recover the original image.

Image Restoration refers to a class of methods that aim to remove or reduce the degradations that have occurred while the digital image was being obtained.
All natural images when displayed have gone through some sort of degradation:

a) During display mode
b) Acquisition mode, or
c) Processing mode

The degradations may be due to

a) Sensor noise
b) Blur due to camera mis focus
c) Relative object-camera motion
d) Random atmospheric turbulence
e) Others

3.1.1 A Model of Image Restoration Process

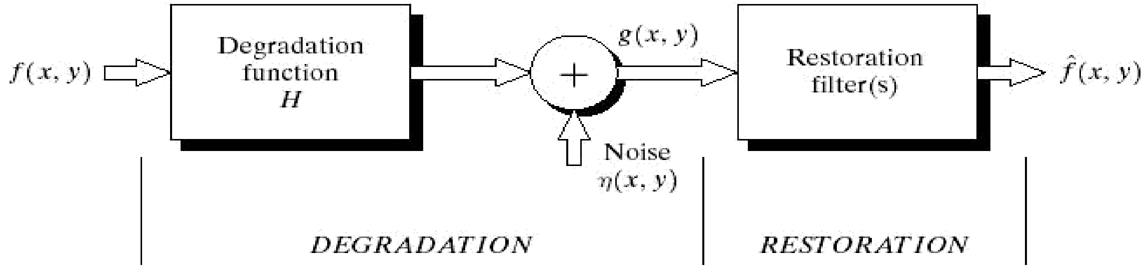
Degradation process operates on a degradation function that operates on an input image with an additive noise term.
Input image is represented by using the notation f(x,y), noise term can be represented as **η**(x,y).These two terms when combined gives the result as g(x,y).
If we are given g(x,y), some knowledge about the degradation function H or J and some knowledge about the additive noise teem **η**(x,y), the objective of restoration is to obtain an estimate *f'(x,y)* of the original image. We want the estimate to be as close as possible to the original image. The more we know about h and **η** , the closer f(x,y) will be to f'(x,y).
If it is a linear position invariant process, then degraded image is given in the spatial domain by

$$g(x,y)=f(x,y)*h(x,y)+\eta(x,y)$$

h(x,y) is spatial representation of degradation function and symbol * represents convolution. In frequency domain we may write this equation as

$$G(u,v)=F(u,v)H(u,v)+N(u,v)$$

The terms in the capital letters are the Fourier Transform of the corresponding terms in the spatial domain.



The image restoration process can be achieved by inversing the image degradation process, i.e.,

$$\hat{G}(u,v) = \frac{F(u,v) - N(u,v)}{H(u,v)} = \frac{F(u,v)}{\hat{H}(u,v)}$$

where $1/\hat{H}(u,v)$ is the inverse filter, and $\hat{G}(u,v)$ is the recovered image. Although the concept is relatively simple, the actual implementation is difficult to achieve, as one requires prior knowledge or identifications of the unknown degradation function $h(x,y)$ and the unknown noise source $n(x,y)$.

In the following sections, common noise models and method of estimating the degradation function are presented.

3.1.2 **Noise Models**

The principal source of noise in digital images arises during image acquisition and /or transmission. The performance of imaging sensors is affected by a variety of factors, such as environmental conditions during image acquisition and by the quality of the sensing elements themselves. Images are corrupted during transmission principally due to interference in the channels used for transmission. Since main sources of noise presented in digital images are resulted from atmospheric disturbance and image sensor circuitry, following assumptions can be made:

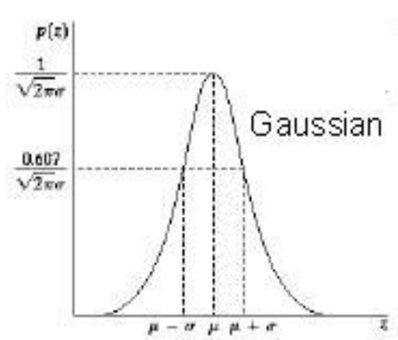
- The noise model is spatial invariant, i.e., independent of spatial location.
- The noise model is uncorrelated with the object function.

I. Gaussian Noise

These noise models are used frequently in practices because of its tractability in both spatial and frequency domain.
The PDF of Gaussian random variable, z is given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

z= gray level
μ= mean of average value of
z **σ**= standard deviation



II. Rayleigh Noise

Unlike Gaussian distribution, the Rayleigh distribution is no symmetric. It is given by the formula.

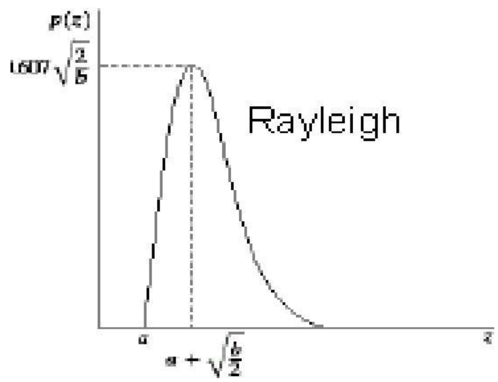
$$p(z) = \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}}, \quad \text{for } z \geq a$$

The mean and variance of this density

Mean/variance

$$\mu = a + \sqrt{\pi b / 4}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}$$



It is displaced from the origin and skewed towards the right.

III. Erlang (gamma) Noise

The PDF of Erlang noise is given by

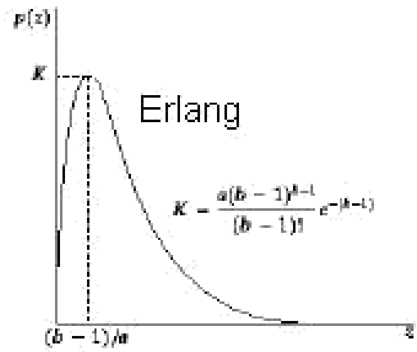
$$p(z) = \frac{a^b z^{b-1}}{(b-1)!}e^{-az}, \quad \text{for } z \geq 0$$

The mean and variance of this noise is

Mean/variance

$$\mu = \frac{b}{a}$$

$$\sigma^2 = \frac{b}{a^2}$$



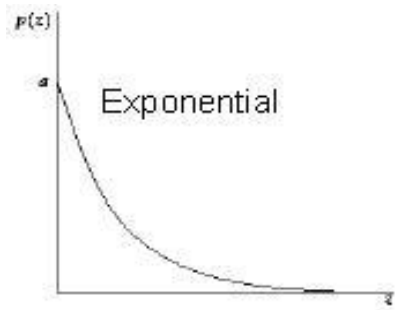
Its shape is similar to Rayleigh disruption.
This equation is referred to as gamma density it is correct only when the denominator is the gamma function.

IV. Exponential Noise

Exponential distribution has an exponential shape.
The PDF of exponential noise is given as

$$p(z) = ae^{-az}, \quad \text{for } z \geq 0$$

Where a>0
It is a special case of Erlang with b=1



V. Uniform Noise

The PDF of uniform noise is given by

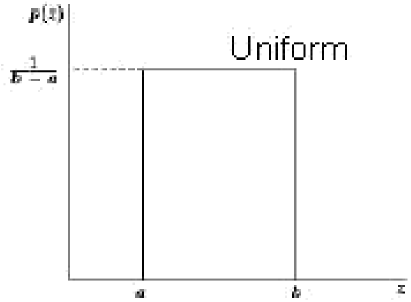
$$p(z) = \begin{cases} \dfrac{1}{(b-a)} & a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

The mean of this density function is given by

Mean/variance

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$

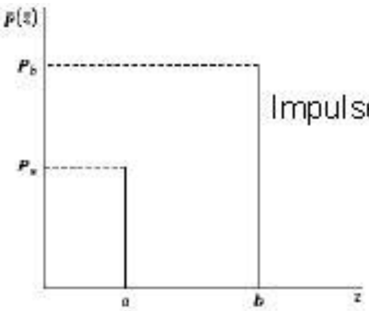


VI. Impulse (Salt and Pepper)Noise

In this case, the noise is signal dependent, and is multiplied to the image.
The PDF of bipolar (impulse) noise is given by

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

If b>a, gray level b will appear as a light dot in image.
Level a will appear like a dark dot.



3.1.3 Restoration In the Presence of Noise Only-Spatial Filtering

When the only degradation present in an image is noise, i.e.

$$g(x,y)= f(x,y)+ \boldsymbol{\eta}(x,y)$$
or
$$G(u,v)= F(u,v)+ N(u,v)$$

The noise terms are unknown so subtracting them from g(x,y) or G(u,v) is not a realistic approach. In the case of periodic noise it is possible to estimate N(u,v) from the spectrum G(u,v). So N(u,v) can be subtracted from G(u,v) to obtain an estimate of original image. Spatial filtering can be done when only additive noise is present.
The following techniques can be used to reduce the noise effect:

3.1.3.1 Mean Filter

3.1.3.1.1 Arithmetic Mean Filter

It is the simplest mean filter. Let S_{xy} represents the set of coordinates in the sub image of size m*n centered at point (x,y). The arithmetic mean filter computes the average value of the corrupted image g(x,y) in the area defined by S_{xy}. The value of the restored image f at any point (x,y) is the arithmetic mean computed using the pixels in the region defined by S_{xy}.

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

This operation can be using a convolution mask in which all coefficients have value 1/mn

A mean filter smoothes local variations in image Noise is reduced as a result of blurring. For every pixel in the image, the pixel value is replaced by the mean value of its neighboring pixels $(N\times M)$ with a weight $w_k=1/(NM)$. This will resulted in a smoothing effect in the image.

3.1.3.1.2 Geometric mean filter

An image restored using a geometric mean filter is given by the expression

$$\hat{f}(x,y) = \left(\prod_{(s,t)\in S_{xy}} g(s,t) \right)^{1/mn}$$

Here, each restored pixel is given by the product of the pixel in the subimage window, raised to the power 1/mn. A geometric mean filters but it to loose image details in the process.

3.1.3.1.3 Harmonic mean filter

The harmonic mean filtering operation is given by the expression

$$\hat{f}(x,y) = \sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1} \Big/ \sum_{(s,t)\in S_{xy}} g(s,t)^{Q}$$

The harmonic mean filter works well for salt noise but fails for pepper noise. It does well with Gaussian noise also.

3.1.3.1.4 Order statistics filter

Order statistics filters are spatial filters whose response is based on ordering the pixel contained in the image area encompassed by the filter.

The response of the filter at any point is determined by the ranking result.

3.1.3.1.4.1 Median filter

It is the best order statistic filter; it replaces the value of a pixel by the median of gray levels in the Neighborhood of the pixel.

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\mathbf{median}} \{g(s,t)\}$$

The original of the pixel is included in the computation of the median of the filter are quite possible because for certain types of random noise, the provide excellent noise reduction capabilities with considerably less blurring then smoothing filters of similar size. These are effective for bipolar and unipolor impulse noise.

3.1.3.1.4.1 Max and Min Filters

Using the l00th percentile of ranked set of numbers is called the max filter and is given by the equation

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\} \quad .$$

It is used for finding the brightest point in an image. Pepper noise in the image has very low values, it is reduced by max filter using the max selection process in the sublimated area sky.

The 0th percentile filter is min filter

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

This filter is useful for flinging the darkest point in image. Also, it reduces salt noise of the min operation.

a. Midpoint Filter

The midpoint filter simply computes the midpoint between the maximum and minimum values in the area encompassed by the filter

$$\hat{f}(x,y) = \left(\max_{(s,t)\in S_{xy}} \{g(s,t)\} + \min_{(s,t)\in S_{xy}} \{g(s,t)\} \right) \Big/ 2$$

It comeliness the order statistics and averaging .This filter works best for randomly distributed noise like Gaussian or uniform noise.

3.1.4 **Periodic Noise By Frequency Domain Filtering**

These types of filters are used for this purpose-

3.1.4.1 Band Reject Filters

It removes a band of frequencies about the origin of the Fourier transformer.

3.1.4.1.1 Ideal Band reject Filter

An ideal band reject filter is given by the expression

$$H(u,v) = \begin{cases} 1 & \text{if} \quad D(u,v) < D_0 - W/2 \\ 0 & \text{if} \quad D_0 - W/2 \le D(u,v) \le D_0 + W/2 \\ 1 & \text{if} \quad D(u,v) > D_0 + W/2 \end{cases}$$

D(u,v)- the distance from the origin of the centered frequency rectangle.
W- the width of the band
Do- the radial center of the frequency rectangle.

3.1.4.1.2 Butterworth Band reject Filter

$$H(u,v) = 1 \Big/ \left[1 + \left(\frac{D(u,v)W}{D^2(u,v) - D_0^2} \right)^{2n} \right]$$

3.1.4.1.3 Gaussian Band reject Filter

$$H(u,v) = 1 - \exp\left[-\frac{1}{2} \left(\frac{D^2(u,v) - D_0^2}{D(u,v)W} \right)^2 \right]$$

These filters are mostly used when the location of noise component in the frequency domain is known. Sinusoidal noise can be easily removed by using these kinds of filters because it shows two impulses that are mirror images of each other about the origin. Of the frequency transform.

3.1.4.2 Band Pass Filters

The function of a band pass filter is opposite to that of a band reject filter It allows a specific frequency band of the image to be passed and blocks the rest of frequencies.
The transfer function of a band pass filter can be obtained from a corresponding band reject filter with transfer function $H_{br}(u,v)$ by using the equation-

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$

These filters cannot be applied directly on an image because it may remove too much details of an image but these are effective in isolating the effect of an image of selected frequency bands.

3.1.5 **Notch Filters**

This type of filters rejects frequencies I predefined in neighborhood above a centre frequency These filters are symmetric about origin in the Fourier transform the transfer function of ideal notch reject filter of radius do with centre at (…) and by symmetry at (……) is

$$H(u,v) = \begin{cases} 0 & \text{if} \quad D_1(u,v) \le D_0 \text{ or } D_2(u,v) \le D_0 \\ 1 & \text{otherwise} \end{cases}$$

Where

$$D_1(u,v) = \sqrt{(u - M/2 - u_0)^2 + (v - N/2 - v_0)^2}$$
$$D_2(u,v) = \sqrt{(u - M/2 + u_0)^2 + (v - N/2 + v_0)^2}$$

Butterworth notch reject filter of order n is given by

$$H(u,v) = 1 - \exp\left[-\frac{1}{2}\left(\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right)\right]$$

A Gaussian notch reject filter has the fauna

$$H(u,v) = 1 \Big/ \left[1 + \left(\frac{D_0^2}{D_1(u,v)D_2(u,v)}\right)^n\right]$$

These filter become high pass rather than suppress. The frequencies contained in the notch areas.
These filters will perform exactly the opposite function as the notch reject filter.
The transfer function of this filter may be given as

$$Hnp(u,v) = 1 - Hnr(u,v)$$

$H_{np}(u,v)$- transfer function of the pass filter
$H_{nr}(u,v)$- transfer function of a notch reject filter

3.1.6 **Minimum Mean Square Error (Wiener) Filtering**

This filter incorporates both degradation function and statistical behavior of noise into the restoration process.
The main concept behind this approach is that the images and noise are considered as random variables and the objective is to find an estimate f of the uncorrupted image f such that the mean sequence error between then is minimized.

$$\hat{f}(x) = \sum_{s=-\infty}^{\infty} h_W(x-s)g(s),$$

This error measure is given by

$$e^2 = E\{[f(x) - \hat{f}(x)]^2\} = \min$$

Where e( ) is the expected value of the argument
Assuming that the noise and the image are uncorrelated (means zero average value) one or other has zero mean values
The minimum error function of the above expression is given in the frequency …….. is given by the expression.

$$H_W(u,v) = \frac{H^*(u,v)S_{ff}(u,v)}{|H(u,v)|^2 S_{ff}(u,v) + S_{nn}(u,v)} = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_{nn}(u,v)/S_{ff}(u,v)}$$

Product of a complex quantity with its conjugate is equal to the magnitude of …… complex quantity squared. This result is known as wiener Filter The filter was named so because of the name of its inventor N Wiener. The term in the bracket is known as minimum mean square error filter or least square error filter.
H*(u,v)-degradation function .
H*(u,v)-complex conjugate of H(u,v)
H(u,v) H(u,v)
Sn(u,v)=IN(u,v)I^2- power spectrum of the noise
Sf(u,v)=IF(u,v)^2- power spectrum of the underrated image
H(u,v)=Fourier transformer of the degraded function
G(u,v)=Fourier transformer of the degraded image
The restored image in the spatial domain is given by the inverse Fourier transformed of the frequency domain estimate F(u,v).
Mean square error in statistical form can be approveiment by the function

$$H_W(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}$$

3.1.7 **Inverse Filtering**

It is a process of restoring an image degraded by a degradation function H. This function can be obtained by any method.
The simplest approach to restoration is direct, inverse filtering.
Inverse filtering provides an estimate F(u,v) of the transform of the original image simply by during the transform of the degraded image G(u,v) by the degradation function.

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

It shows an interesting result that even if we know the depredation function we cannot recover the underrated image exactly because N(u,v) is not known .
If the degradation value has zero or very small values then the ratio N(u,v)/H(u,v) could easily dominate the estimate F(u,v).

UNIT-4(IMAGE COMPRESSION)

1. INTRODUCTION

Image: An image is the result of a visual signal of any object captured by optical devices such as human eyes, camera lenses etc., Image is the two dimensional representation of the object.

Digital image: Image when stored in electronic devices such as computers, it is known as the digital image. Thus a digital image is the numeric representation of an image. Digital images are of two types – raster and vector. Raster images refers to the images built using a finite set of digital values, called picture elements or pixels. Pixel is the smallest element in an image. The value that is equivalent to the colour of that specific pixel is stored in the place of the pixel. Images stored in electronic devices fall under this category. Vector images refer to the images that are resulted from mathematical calculations. Vector refers to the elements that have both magnitude and direction (Raster images have only magnitude). In electronic devices, vector type is used to refer texts stored as images or to store images that require more accuracy even in zoomed level. Vectors have both magnitude and direction. So the images are clear even when zoomed. Usually the term digital image refers to raster images.

Need for Compression: The need for storage of digital information increases rapidly with the growth of internet technology. Images are widely used in applications like medical, army and satellite images. Everyday an enormous amount of information is stored, processed and transmitted digitally [4]. The unprocessed or raw digital images occupy more space. This necessitates compression of data. Image compression is the process of representing the required information with the possible minimal amount of data. Hence it is useful for reducing the amount of memory space need for the file. Thus for the growing requirements of image storage and digital technology, storing the compressed images allows us to store more images in the

given disk space. Also transmission rate of images gets increased due to compression. This paper deals with the process of compression in the second part, Compression types in the Third part, Compression techniques in the Fourth part and gets concluded in the fifth part.

2. COMPRESSION PROCESS Compression

Compression is achieved by removing one or more of the three basic data redundancies. They are

- Coding Redundancy
- Interpixel Redundancy
- Psychovisual Redundancy.

Data redundancy refers to the data that are in the state of being not or no longer needed.

2.1 Coding Redundancy

A code is a set of symbols that represents some information in the image. As a method of compression, codes occurring many times are represented with lesser bits and other codes are represented with more number of bits. In that, coding redundancy is the form of data redundancy that is present when less than optimal code words are used to represent any information[1] and so this redundancy deals with the data representation of the image.

2.2 Interpixel Redundancy:

Interpixel redundancy is the form of data redundancy that is related to interpixel correlation within an image [11]. While storing an image, it is not necessary to store all the details of pixels in the image. This is allowed due to the reason that the value of any pixel can be calculated from the values of the neighboring pixels. This is known as interpixel dependency. Interpixel correlation is the difference between the information content and the information capacity of the image. In an image, consequent pixels may contain same colour or relative shades of same colfour. Instead storing the full details of all the pixels the values of correlated pixels may be calculated from the details of a single pixel for which full details are stored. Thus this data redundancy works on image representation.

Let us consider any two pixels which are represented as functions f(r) and g(r) respectively. The formula used to find the correlation between those two pixels is given by the convolution theorem,

Depending upon the value of the equation (B) we can say that the two pixels are correlated or not. A Threshold T is fixed and if the value of (B) is greater than T then the pixels are not correlated else the pixels are correlated. If the pixels
are auto-correlated then we have . The value of correlated pixels can be calculated based on two operations namely repetition and prediction. Repetition is used for auto-correlated pixels and prediction is used for other correlated pixels. This means the relatively or equally valued pixels are calculated. This method can be applied for both colour and grayscale images. Interpixel redundancy is of two types – spatial and Interframe.

Spatial Redundancy: Spatial redundancy refers to redundancy in still images. Consider any pixel, A having coordinates (x,y), then Value(x,y) always depends on (x',y') and the Value(x',y'), for all (x',y') belongs to the neighbourhood of (x,y).

Interframe redundancy: Interframe redundancy refers to the redundancy present in videos. Consider the set of frames (x,y,t_i), for all i=0,1,2,… then Value(x,y,t_i) is related to each other.

Interpixel redundancy is removed using compression techniques like Constant area coding, Run length encoding.

2.3 Psychovisual Redundancy:

Psychovisual redundancy is the data redundancy that works with respect to image content. The psychovisual redundancy refers to the information that is not noticed or processed by the human eyes. This is because certain information has more importance for Human Visual System(HVS) than others based upon the fact that Human eye has the effect of

recognizing only low frequencies and neglecting high frequencies. Thus its function as a low pass filter is used by compression algorithms, to reduce the probably high frequency signal.

By removing the psychovisually redundant information, the compression achieved is lossy but the end user who is a human being, cannot find the difference.

To find the psychovisually redundant information, a discrimination threshold ( ) is found for all the pixels and this value is added to the initial value of the pixel . After that, the pixel having smallest value is the psychovisually redundant information.

Discrimination threshold is calculated using the Weber's Law

where K_w is a constant known as the Weber's fraction. i.e the value of always depends upon the .

The equation (C) may be rewritten as
which is similar to the equation of a straight line
with K_w as the slope and b value as 0. Thus we can see all
the values of lies in a straight line when plotted using a
graph. Lossy compression algorithms like JPEG compression works to remove this type of redundancy.

3. COMPRESSION TYPES

Compression process always results in encoding of the input. Then to retrieve the original image decompression is used. Based on whether the original image can be retrieved from the compressed file, the types of compression are

1. Lossless compression
2. Lossy compression

3.1 Lossless Compression Technique:

In lossless compression technique, the original image is perfectly retrieved from the compressed image. This is also known as noiseless coding. The lossless compression techniques follow the entropy encoding schemes. In entropy encoding, the nature of the input data is analyzed and the encoding or further process is performed based on it.

3.1.1 Basic Steps in Lossless Compression Technique:

Lossless compression is done in following steps:

1. Analysis and grouping of data.
2. Encoding.

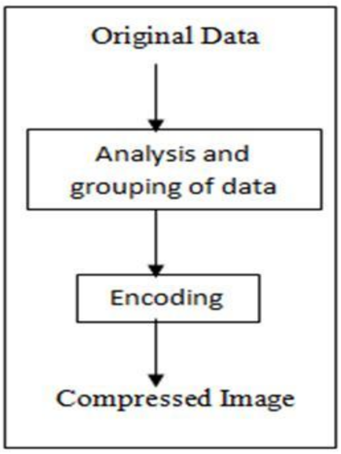


Fig-1: Lossless compression process

Matrix transformation: Matrix transformation is performed for the transformation of input data to the required approach. Algorithms like DCT is used in this step. The output of this algorithm is lossless when compared with its input.

Quantization: Quantization is the rounding – off process. Quantization is a step that has its results as lossy when
compared with its input. Quantization process is irreversible. So it is not present in decompression.

Encoding: This step encodes the quantized input to be suitable to store. The output of this step is compressed when compared with its input. Encoding is the key step in any compression process. Other steps are preprocesses to encoding.

Analysis and Grouping of data:
The input data is analyzed and grouped together based on the compression technique involved.

Encoding:
The compression technique is applied in this step to the previously grouped or analyzed data.

3.2 Lossy Compression Technique

In lossy compression technique the original images are not exactly reconstructed after decompression. But the output produced will be reasonably close to the original image. Only data that is considered trivial is deleted. The remaining data will be sufficient to represent the required information. Also in lossy compression schemes there are more possibilities for higher compression ratios. The lossy compression techniques follow source encoding schemes. In source encoding, the nature of the input data is not needed to be analyzed.

3.2.1 Basic Steps in Lossy Compression:

Any lossy compression process is done in three steps:

1. Matrix Transformation
2. Quantization
3. Encoding

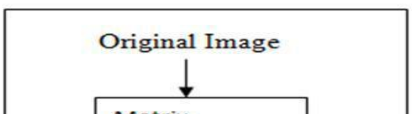


Fig-2: Lossy compression process

4. COMPRESSION TECHNIQUES

There are a large number of encoding techniques separately for lossy and lossless compressions, in which, the following are most important and used in many compression algorithms

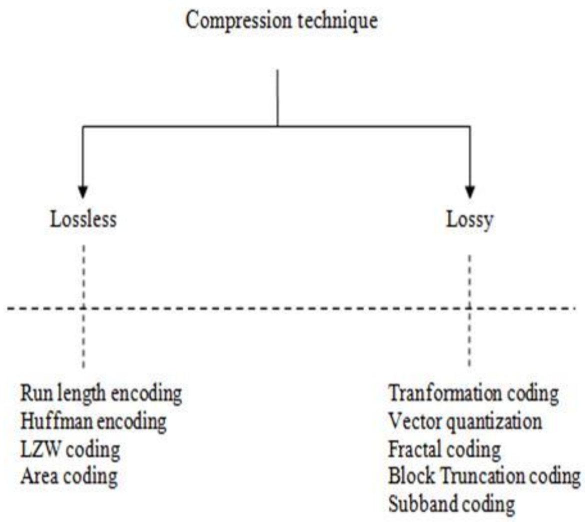


Fig-3: Lossless and Lossy compression techniques

4.1 Lossless Compression Techniques

4.1.1 Run Length Encoding

Run Length Encoding (RLE) is an entropy encoding compression technique that works on interpixel redundancy. This compression proceeds by first, finding the runs of data in the image. Runs of data refer to sequences in which same value occurs in many consecutive elements [16]. When such runs of data are identified, they are stored as a set of two values – one value being the original value that composes the run and the number of times the value is repeated. This compression algorithm is suitable for line drawings, logos and small animation files. For example consider the following example of a binary image of ‘I’ in black, with a white background. This is a stored in the memory as zeros and ones. 0 for white and 1 for black.

4.1.2 Huffman Encoding

This is an entropy encoding technique that works on coding redundancy. This data compression algorithm encodes the data based on their probability of occurrence in the image. The elements of the image that is, the pixels are considered as symbols. All the images that contain the repetition of any particular value, does not contain it as runs. For the compression of those images, Huffman Coding is suitable rather than Run Length Encoding. For symbols that occur more frequently, smaller number of bits are assigned and others are assigned with relatively larger number of bits.
For example, in a text file we consider five symbols: A,E,I,O,U with their probability of occurrence and number of bits allotted for each of them:

Table 1: Bit allocation for various symbols

Symbols	Frequency of occurrence	Numberof bits allotted to encode
A	12	6
E	30	2
I	6	7
O	17	5
U	22	3

4.1.3 LZW Coding:

This entropy encoding technique works in the way that, various patterns in the image are noted in a dictionary and then the original image is encoded by linking the pattern in the dictionary. Once a pattern is created, whenever any pattern is encountered, the algorithm checks for whether that pattern is present already in the dictionary. If yes, then a link to that pattern in the dictionary is given. If no then that pattern is included in the dictionary and then it is linked in the original file. Hence this algorithm is known as “dictionary based compression algorithm”. This encoding scheme is optimal for images having a large number of repeated patterns and not suitable for images having less number of repeated patterns since the dictionary occupies a large amount of memory space which in turn makes the file bigger.

For example Consider the occurrence of patterns “en, op, ng, co, tr, to, di” , in the text “entzropy encoding”

Table 2: Dictionary of patterns

1	**2**	**3**	**4**	**5**	**6**	**7**
en	**Op**	**Ng**	**Co**	**Tr**	**to**	**di**

The encoded text is: “152y 1473”. The size of a pattern is taken as two. While grouping the texts, when a single element resides unable to be grouped, then that element is written as it is.

4.1.4 Area Coding

Area coding is an entropy encoding scheme which works on binary images. This works in the way that the whole image is divided into some blocks of same size so the resulting block may be of three types – blocks having only white pixels, blocks having only black pixels and blocks having both black and white pixels. Then the block that occurs frequently, is assigned with a 1-bit code word and the others are assigned with 2-bit code words. Thus by replacing blocks with 1-bit or 2-bit code words the file size gets reduced and thus the compression is achieved.

4.2 Lossy Compression Techniques

4.2.1 Transformation Coding

This source encoding scheme works in such a way that, the pixels in the original image is transformed into frequency domain coefficients. These frequency domain coefficients are known as transform coefficients. These coefficients have certain useful properties. For example: it has the energy compaction property due to which maximum amount of energy in the original data gets concentrated in few of the transform coefficients. These coefficients alone are selected, the remaining are deleted to achieve compression. To the selected coefficients, further processing is applied.DCT coding has been the most common approach to perform transform coding. It is also adopted in the JPEG image compression standard.

4.2.2 Vector Quantization

This source encoding technique is otherwise known as block quantization or pattern matching quantization[16]. The input image which contains various amplitude levels is divided into various blocks. The amplitude levels present in the input are in the form of vectors since they have both magnitude and direction. Consider an i-dimensional vector

. It is matched to a of j-dimensional vectors with $j < i$. Vector space y contains the quantized values of the vector space x. In quantization phase, the vectors are quantized in whole as a block instead of quantizing each sample. Thus this has the density matching property due to which some errors occur and these errors are inversely proportional to the density of the blocks. The quantized values of the vector space are stored as code words in a codebook. Only the index of the codeword in the codebook is used and thus compression is achieved through this also.

4.2.4 Block Truncation Coding:

In this source encoding technique, image is divided into non overlapping blocks of pixels. Threshold value is determined for every block. Here the threshold value is the mean of the values in the block. It is denoted assegments. Each segment now contains zeros and ones. Reconstruction value of these blocks is found out. Reconstruction is made using two values, k and l

4.2.5 Subband Coding

In this source encoding technique, the input frequency band is divided into different sub-bands using digital filter bank.

This digital filtration is based on the separation of low and high frequencies. Thus these filters act as the low pass and high pass filters. For further separation, each of the sub-bands is applied with low pass and high pass filters. This compression technique works to remove the psychovisually redundant information and hence some of the high frequency data is removed. Then each of the sub-bands is quantized separately. As the final step, the encoding of the bands is performed. The encoding technique that is suitable for each of the sub-bands is applied without being interrupted by others.

Wavelet Coding:

The wavelet coding is based on the idea that the coefficients of a transform that decorrelates the pixels of an image can be coded more efficiently than the original pixels themselves. If the transform's basis functions—in this case wavelets—pack most of the important visual information into a small number of coefficients, the remaining coefficients can be quantized coarsely or truncated to zero with little image distortion.

Figure 11 shows a typical wavelet coding system. To encode a 2^J X 2^J image, an analyzing wavelet, Ψ, and minimum decomposition level, J - P, are selected and used to compute the image's discrete wavelet transform. If the wavelet has a complimentary scaling function φ, the fast wavelet transform can be used. In either case, the computed transform converts a large portion of the original image to horizontal, vertical, and diagonal decomposition coefficients with zero mean and Laplacian-like distributions.

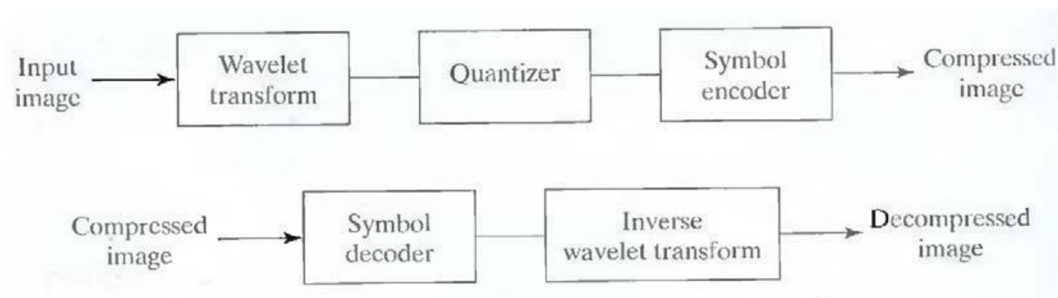


Fig.11 A wavelet coding system: (a) encoder; (b) decoder.

Since many of the computed coefficients carry little visual information, they can be quantized and coded to minimize intercoefficient and coding redundancy. Moreover, the quantization can be adapted to exploit any positional correlation across the P decomposition levels. One or more of the lossless coding methods, including run-length, Huffman, arithmetic, and bit-plane coding, can be incorporated into the final symbol coding step. Decoding is accomplished by inverting the encoding operations—with the exception of quantization, which cannot be reversed exactly.

The principal difference between the wavelet-based system and the transform coding system is the omission of the transform coder's subimage processing stages.

Because wavelet transforms are both computationally efficient and inherently local (i.e., their basis functions are limited in duration), subdivision of the original image is unnecessary.

Transform Coding:

All the predictive coding techniques operate directly on the pixels of an image and thus are spatial domain methods. In this coding, we consider compression techniques that are based on modifying the transform of an image. In transform coding, a reversible, linear transform (such as the Fourier transform) is used to map the image into a set of transform coefficients, which are then quantized and coded. For most natural images, a significant number of the coefficients have small magnitudes and can be coarsely quantized (or discarded entirely) with little image distortion. A variety of transformations, including the discrete Fourier transform (DFT), can be used to transform the image data.

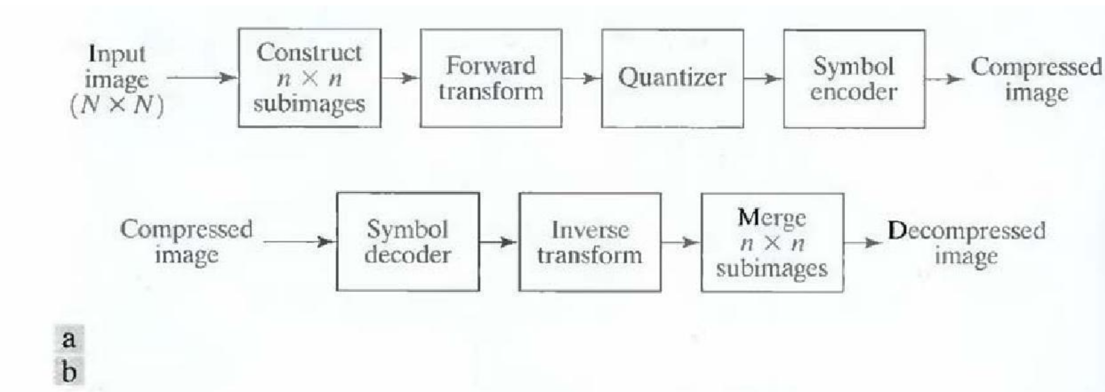


Fig. 10 A transform coding system: (a) encoder; (b) decoder.

Figure 10 shows a typical transform coding system. The decoder implements the inverse sequence of steps (with the exception of the quantization function) of the encoder, which performs four relatively straightforward operations: subimage decomposition, transformation, quantization, and coding. An N X N input image first is subdivided into subimages of size n X n, which are then transformed to generate $(N/n)^2$ subimage transform arrays, each of size n X n. The goal of the transformation process is to decorrelate the pixels of each subimage, or to pack as much information as possible into the smallest number of transform coefficients. The quantization stage then selectively eliminates or more coarsely quantizes the coefficients that carry the least information. These coefficients have the smallest impact on reconstructed subimage quality. The encoding process terminates by coding (normally using a variable-length code) the quantized coefficients. Any or all of the transform encoding steps can be adapted to local image content, called adaptive transform coding, or fixed for all subimages, called nonadaptive transform coding.

Lossy Predictive Coding:

In this type of coding, we add a quantizer to the lossless predictive model and examine the resulting trade-off between reconstruction accuracy and compression performance. As Fig.9 shows, the quantizer, which absorbs the nearest integer function of the error-free encoder, is inserted between the symbol encoder and the point at which the prediction error is formed. It maps the prediction error into a limited range of outputs, denoted $e\^_n$ which establish the amount of compression and distortion associated with lossy predictive coding.

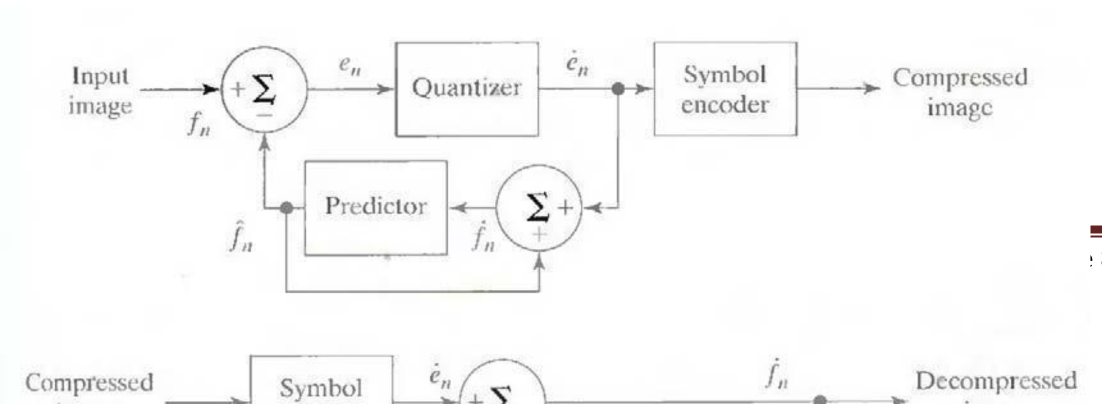


Fig. 9 A lossy predictive coding model: (a) encoder and (b) decoder.

In order to accommodate the insertion of the quantization step, the error-free encoder of figure must be altered so that the predictions generated by the encoder and decoder are equivalent. As Fig.9 (a) shows, this is accomplished by placing the lossy encoder's predictor within a feedback
loop, where its input, denoted $\dot{f}_n$, is generated as a function of past predictions and the corresponding quantized errors. That is,

$$\dot{f}_n = \dot{e}_n + \hat{f}_n$$

This closed loop configuration prevents error buildup at the decoder's output. Note from Fig. 9
(b) that the output of the decoder also is given by the above Eqn.

Optimal predictors:

The optimal predictor used in most predictive coding applications minimizes the encoder's mean-square prediction error

$$E\{e_n^2\} = E\{[f_n - \hat{f}_n]^2\}$$

subject to the constraint that

$$\dot{f}_n = \dot{e}_n + \hat{f}_n \approx e_n + \hat{f}_n = f_n$$

And

$$\hat{f}_n = \sum_{i=1}^{m} \alpha_i f_{n-i}$$

That is, the optimization criterion is chosen to minimize the mean-square prediction error, the quantization error is assumed to be negligible ($\dot{e}_n \approx e_n$), and the prediction is constrained to a linear combination of m previous pixels.1 These restrictions are not essential, but they simplify the analysis considerably and, at the same time, decrease the computational complexity of the predictor. The resulting predictive coding approach is referred to as differential pulse code modulation (DPCM)

Huffman coding:

The most popular technique for removing coding redundancy is due to Huffman (Huffman [1952]). When coding the symbols of an information source individually, Huffman coding yields the smallest possible number of code symbols per source symbol. In terms of the noiseless coding theorem, the resulting code is optimal for a fixed value of n, subject to the constraint that the source symbols be coded one at a time.

The first step in Huffman's approach is to create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into a single symbol that replaces them in the next source reduction. Figure 4.1 illustrates this process for binary coding (K-ary Huffman codes can also be constructed). At the far left, a hypothetical set of source symbols and their probabilities are ordered from top to bottom in terms of decreasing probability values. To form the first source reduction, the bottom two probabilities, 0.06 and 0.04, are combined to form a "compound symbol" with probability 0.1. This compound symbol and its associated probability are placed in the first source reduction column so that the
probabilities of the reduced source are also ordered from the most to the least probable. This process is then repeated until a reduced source with two symbols (at the far right) is reached.

The second step in Huffman's procedure is to code each reduced source, starting with the smallest source and working back to the original source. The minimal length binary code for a two-symbol source, of course, is the symbols 0 and 1. As Fig. 4.2 shows, these symbols are assigned to the two symbols on the right (the assignment is arbitrary; reversing the order of the 0 and 1 would work just as well). As the reduced source symbol with probability 0.6 was generated by combining two symbols in the reduced source to its left, the 0 used to code it is now assigned to both of these symbols, and a 0 and 1 are arbitrarily

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Fig.4.1 Huffman source reductions.

Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Fig.4.2 Huffman code assignment procedure.

appended to each to distinguish them from each other. This operation is then repeated for each reduced source until the original source is reached. The final code appears at the far left in Fig. 4.2. The average length of this code is

$$L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5)$$
$$= 2.2 \text{ bits/symbol}$$

and the entropy of the source is 2.14 bits/symbol. The resulting Huffman code efficiency is 0.973.

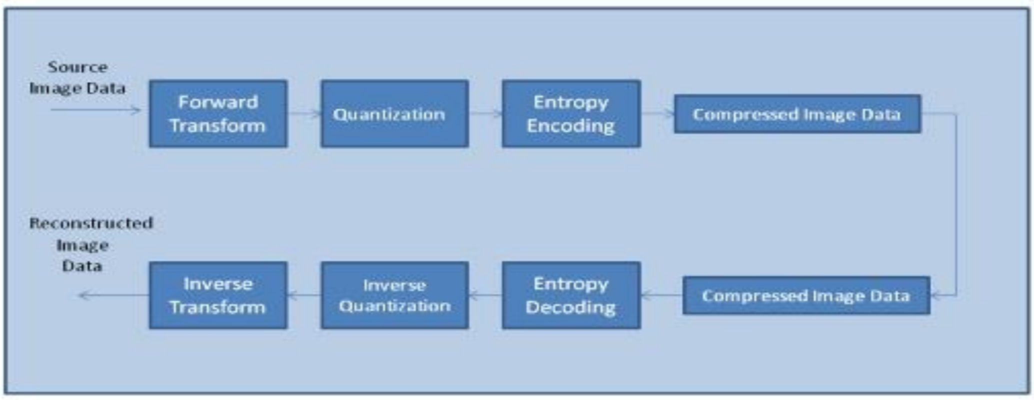
Huffman's procedure creates the optimal code for a set of symbols and probabilities subject to the constraint that the symbols be coded one at a time. After the code has been created, coding and/or decoding is accomplished in a simple lookup table manner. The code itself is an instantaneous uniquely decodable block code. It is called a block code because each source symbol is mapped into a fixed sequence of code symbols. It is instantaneous, because each code word in a string of code symbols can be decoded without referencing succeeding symbols. It is uniquely decodable, because any string of code symbols can be decoded in only one way. Thus, any string of Huffman encoded symbols can be decoded by examining the individual symbols of the string in a left to right manner.

JPEG 2000 STANDARD:-

- Wavelet based image compression standard

Encoding

- Decompose source image into components
- Decompose image and its components into rectangular tiles
- Apply wavelet transform on each tile
- Quantize and collect subbands of coefficients into rectangular arrays of "code-blocks"
- Encode so that certain ROI's can be coded in a higher quality
- Add markers in the bitstream to allow error resilience



Advantages:

- Lossless and lossy compression.
- Progressive transmission by pixel accuracy and resolution.
- Region-of-Interest Coding.
- Random codestream access and processing.
- Robustness to bit-errors.
- Content-based description.
- Side channel spatial information (transparency).

Image sampling and quantization

continuous image (in real life) → digital (computer)

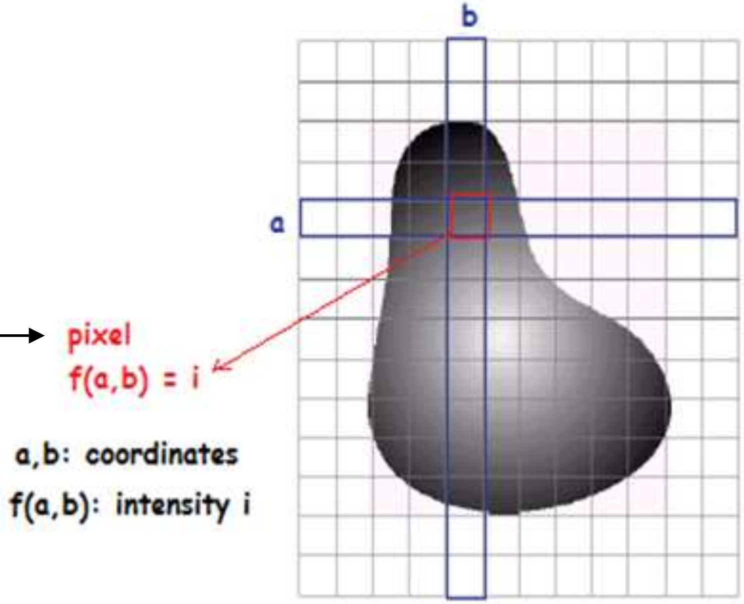
To do this we use Two processes:

sampling and **quantization**.

Remember that:

the image is a function f(x,y),

- x and y are coordinates
- F: intensity value (Amplitude)

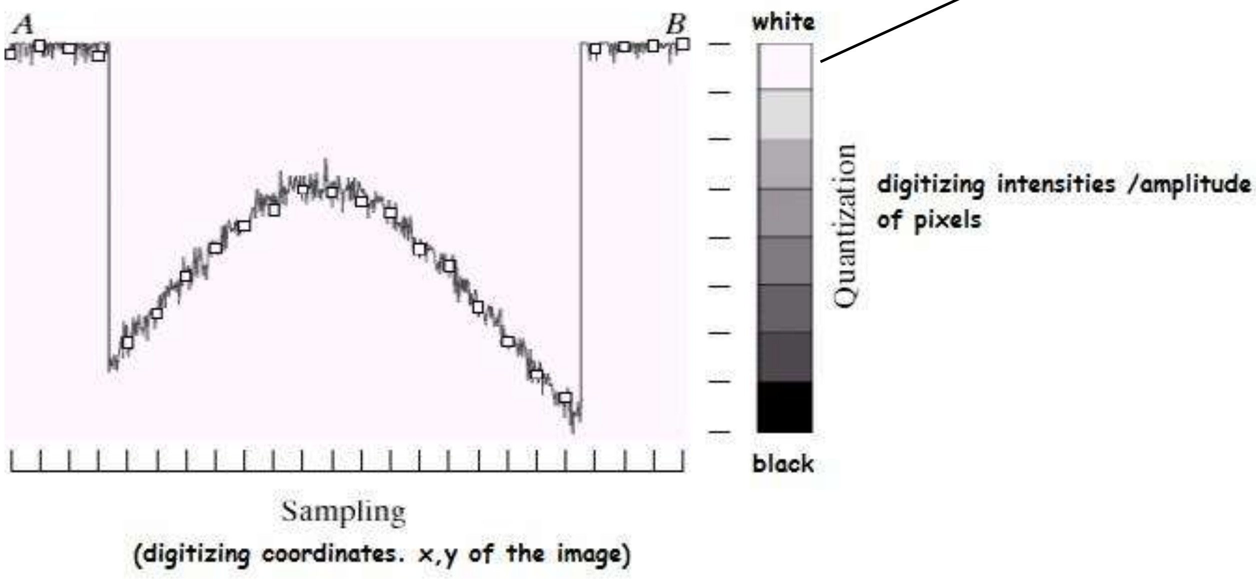


Sampling: digitizing the coordinate values
Quantization: digitizing the amplitude values
Thus, when x, y and f are all finite, discrete quantities, we call the image a digital image.

How does the computer digitize the continuous image?

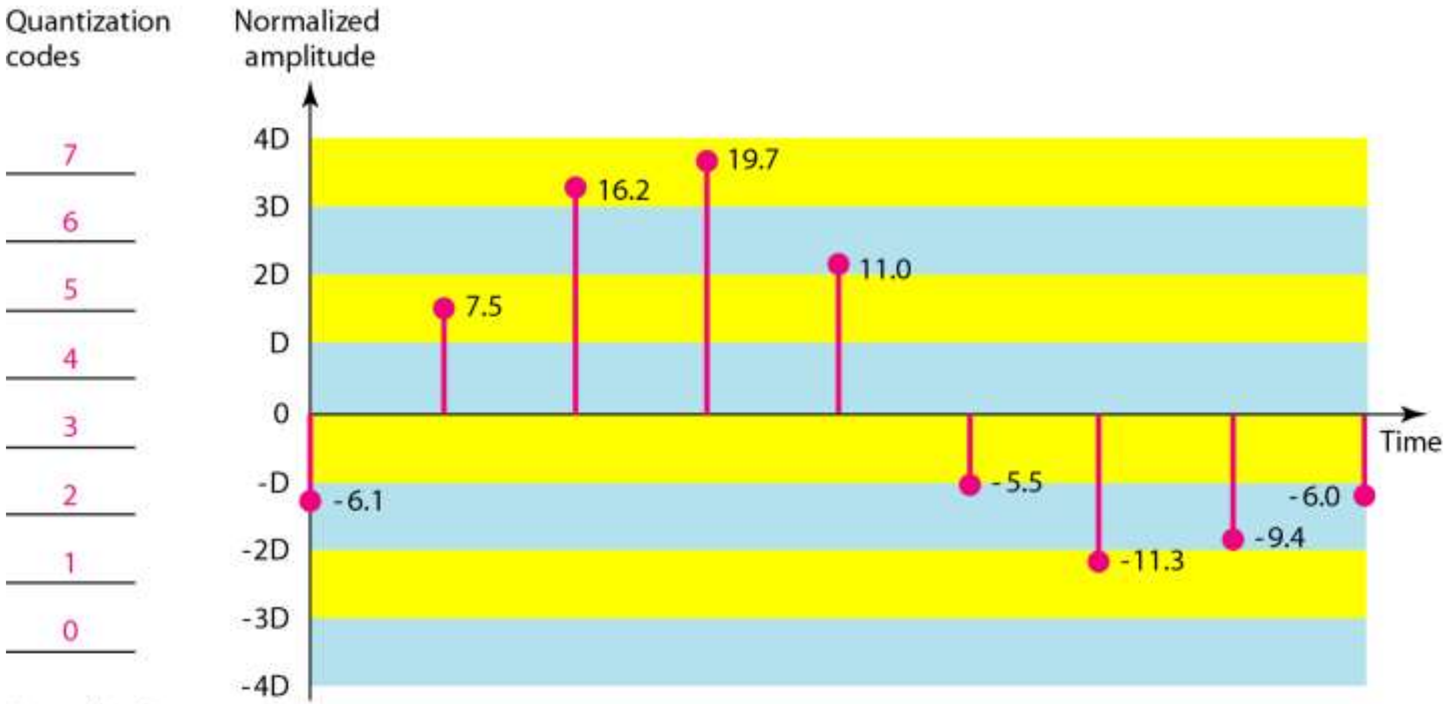
Sampling: digitizing coordinates
Quantization: digitizing intensities

Gray-level scale that divides gray-level into 8 discrete levels



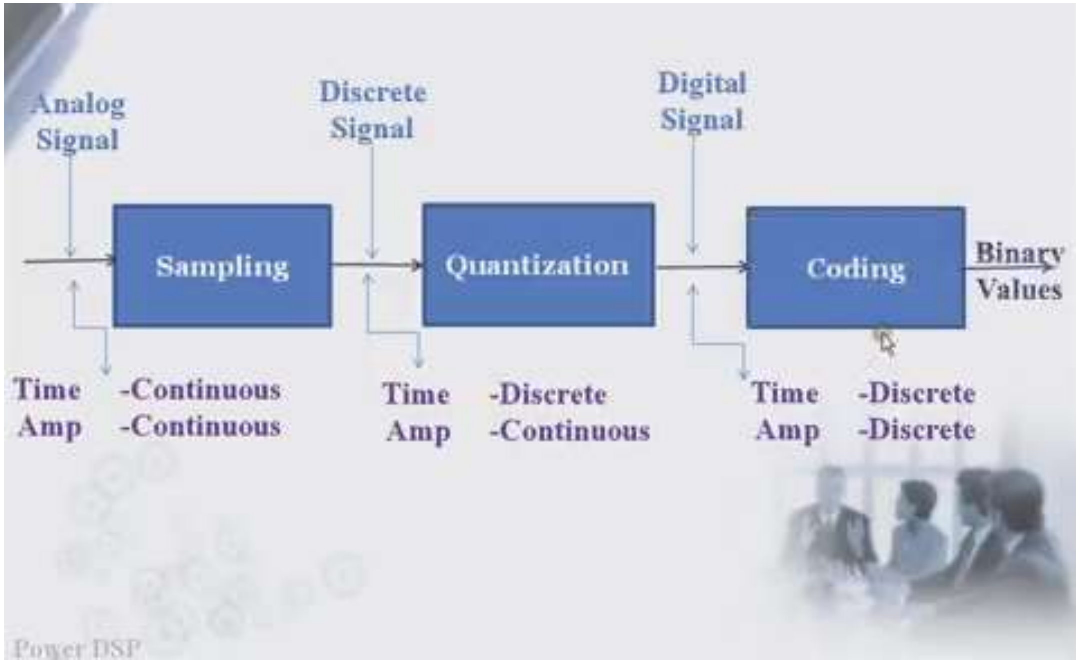
Quantization: converting each sample gray-level value into discrete digital quantity.

sample is a small white square, located by a vertical tick mark as a point x,y



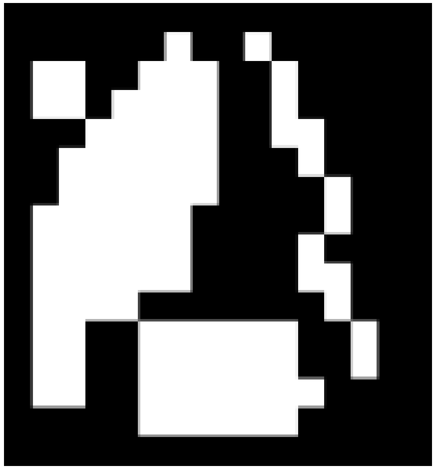
Normalized PAM values	-1.22	1.50	3.24	3.94	2.20	-1.10	-2.26	-1.88	-1.20
Normalized quantized values	-1.50	1.50	3.50	3.50	2.50	-1.50	-2.50	-1.50	-1.50
Normalized error	-0.38	0	+0.26	-0.44	+0.30	-0.40	-0.24	+0.38	-0.30
Quantization code	2	5	7	7	6	2	1	2	2
Encoded words	010	101	111	111	110	010	001	010	010

- A **sample** is a value or set of values at a point in time and/or space.
- **Sampling rate-**A commonly seen measure of sampling is S/s, which stands for "Samples per second." As an example, 1 MS/s is one million samples per second.

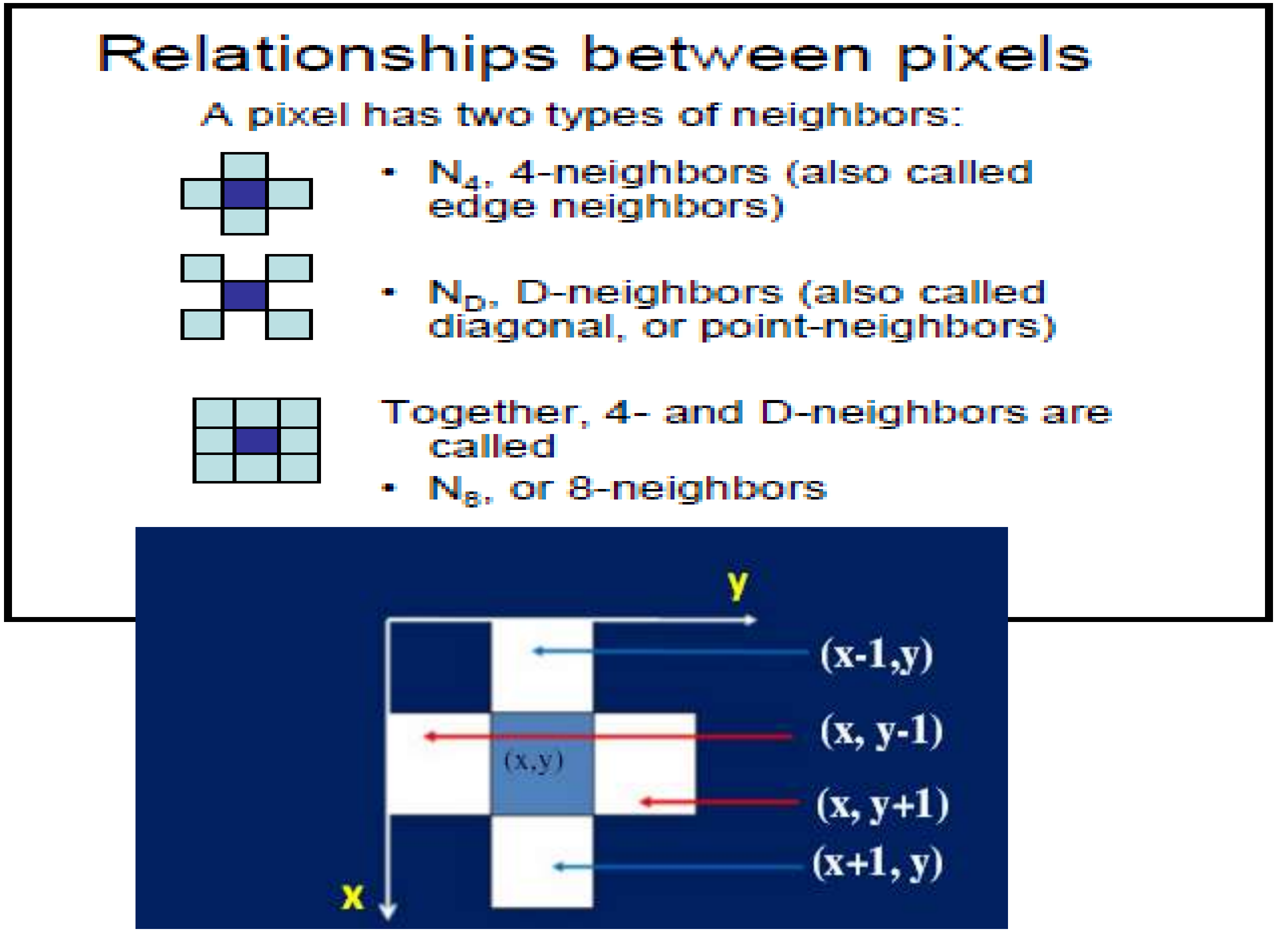


Important relationships between pixels in a digital image.

Why is this important?



How many objects are there in this image?



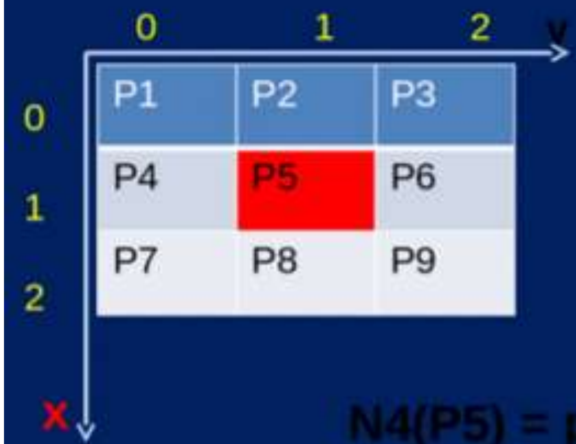
- A pixel p at (x,y) has 2 horizontal and 2 vertical neighbors:
  - (x+1,y), (x-1,y), (x,y+1), (x,y-1)
  - This set of pixels is called the 4-neighbors of p: $N_4(p)$

Neighbors of a Pixel (Contd..)

N_D	N_4	N_D
N_4	P	N_4
N_D	N_4	N_D

- N_4 - 4-neighbors
- N_D - diagonal neighbors
- N_8 - 8-neighbors (N_4 U N_D)

Example



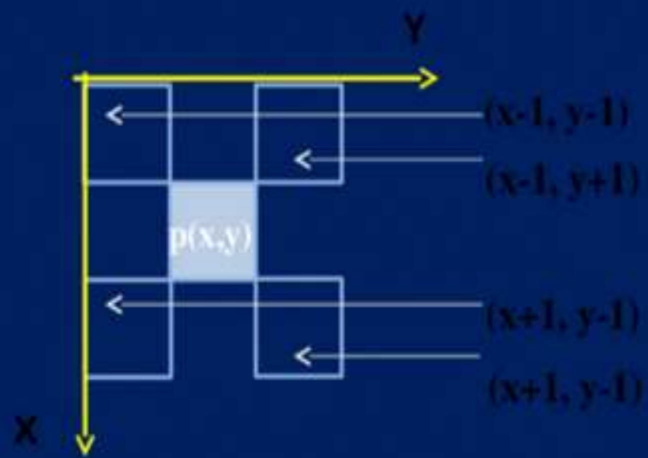
N4(P5) = p4,p6 ---- Horizontal Neighbors

p2,p8 ---- Vertical Neighbors

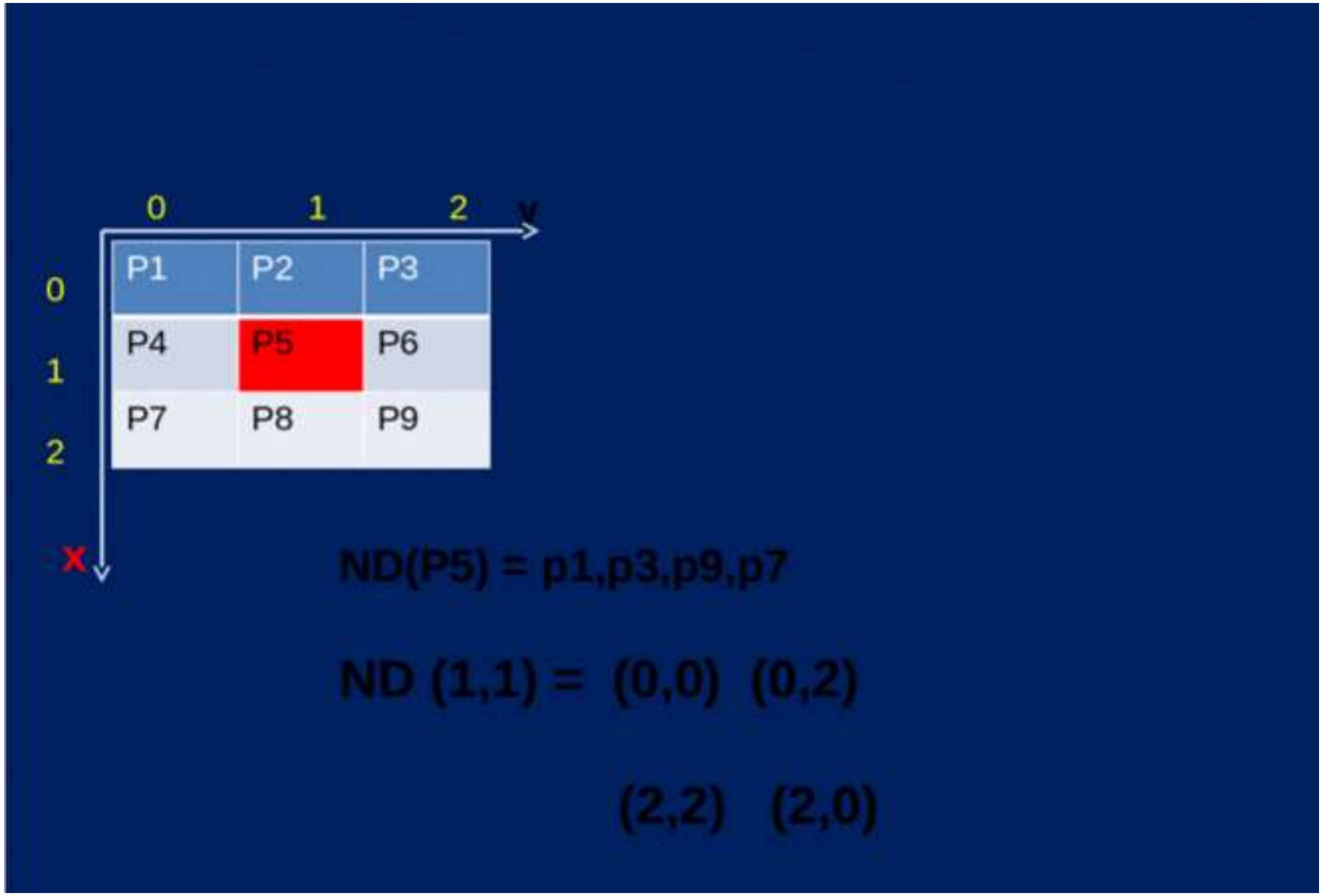
N4 (1,1) = (1,0) (1,2)

Diagonal Neighbours **ND(p)**

- A pixel p at coordinates (x,y) has 4 diagonal neighbours, each being a **unit distance** from (x,y)



Diagonal neighbours



8 Neighbours **N8(p)**

- 8-neighbours of p= 4 diagonal neighbours of p and 4-neighbours of p

N8(p) = ND(p) + N4(p)

Adjacency

- Two pixels are **connected** if they are neighbors and their gray levels satisfy some specified criterion of similarity.
- For example, in a binary image two pixels are connected if they are 4-neighbors and have same value (0/1)
- **Let v:** a set of intensity values used to ***define adjacency*** and ***connectivity***.
- In a **binary Image** **v={1},** if we are referring to adjacency of pixels with value 1.
- In a **Gray scale image**, the idea is the same, but **v** typically contains more elements, for example **v= {180, 181, 182,....,200}.**
- If the possible intensity values 0 to 255, **v** set could be any subset of these 256 values.

Types of adjacency

1. **4-adjacency:** Two pixels **p** and **q** with values from v are **4-adjacent** if **q** is in the set N_4 **(p)**.
2. **8-adjacency:** Two pixels **p** and **q** with values from v are **8-adjacent** if **q** is in the set N_8 **(p).**
3. **m-adjacency (mixed):** two pixels **p** and **q** with values from v are **m-adjacent** if:
   - **q** is in N_4 **(p) or**
   - **q** is in N_D **(P) and**
   - **The set** $N_4(p) \cap N_4(q)$ has no pixel whose values are from v (**No intersection**).
   - **Mixed adjacency** is a modification of 8-adjacency "introduced to eliminate the ambiguities that often arise when 8- adjacency is used. (eliminate multiple path connection)

__Adjacency:__ Two pixels are adjacent if they are neighbors and their intensity level 'V' satisfy some specific criteria of similarity.

e.g. V = {1}
V = { 0, 2}
Binary image = { 0, 1}
Gray scale image = { 0, 1, 2, ------, 255}

In binary images, 2 pixels are adjacent if they are neighbors & have some intensity values either 0 or 1.

In gray scale, image contains more gray level values in range 0 to 255.

4-adjacency: Two pixels p and q with the values from set 'V' are 4-adjacent if q is in the set of $N_4(p)$.

e.g. V = { 0, 1}

1	1	0
1	1	0
1	0	1

p in RED color
q can be any value in GREEN color.

8-adjacency: Two pixels p and q with the values from set 'V' are 8-adjacent if q is in the set of $N_8(p)$.

e.g. V = { 1, 2}

0	1	1
0	2	0
0	0	1

p in RED color
q can be any value in GREEN color

Connectivity: 2 pixels are said to be connected if their exists a path between them.

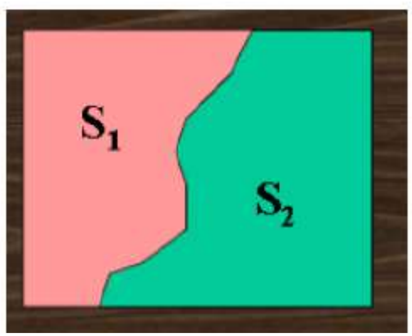
Let 'S' represent subset of pixels in an image.

Two pixels p & q are said to be connected in 'S' if their exists a path between them consisting entirely of pixels in 'S'.

For any pixel p in S, the set of pixels that are connected to it in S is called a **connected component of S**.

Connectivity

- Let **S** represent a subset of pixels in an image. Two pixels **p** and **q** are said to be connected in **S** if there exists a path between them.
- Two image subsets **S1** and **S2** are adjacent if some pixel in **S1** is adjacent to some pixel in **S2**



Paths

Paths: A path from pixel p with coordinate (x, y) with pixel q with coordinate (s, t) is a sequence of distinct sequence with coordinates $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ where

$(x, y) = (x_0, y_0)$
& $(s, t) = (x_n, y_n)$

Closed path: $(x_0, y_0) = (x_n, y_n)$

Paths

<u>Example # 1</u>: Consider the image segment shown in figure. Compute length of the ***shortest-4, shortest-8 & shortest-m paths*** between pixels p & q where,
V = {1, 2}.

4	2	3	2 q
3	3	1	3
2	3	2	2
p 2	1	2	3

<u>Example # 1</u>:

<u>Shortest-4 path:</u>

V = {1, 2}.

4	2	3	2 q
3	3	1	3
2	3	2	2
p 2 →	1	2	

<u>Example # 1</u>:

<u>Shortest-4 path:</u>

V = {1, 2}.

4	2	3	2 q
3	3	1 ↑	3
2	3	2 ↑	2
p 2 →	1 →	2	3

<u>Example # 1</u>:

<u>Shortest-4 path:</u>

V = {1, 2}.

4	2	3 ×	2 q
3	3	1 ↑	3
2	3	2 ↑	2
p 2 →	1 →	2	3

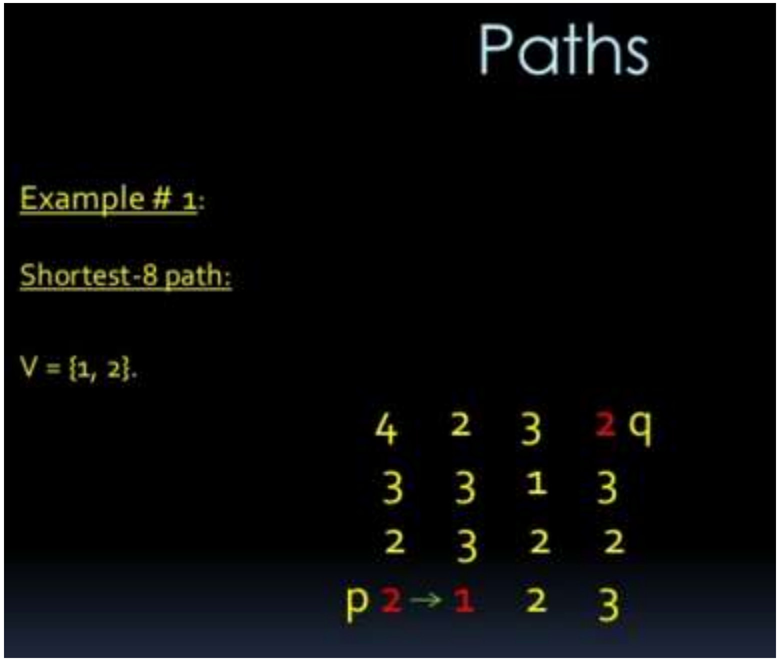
Paths

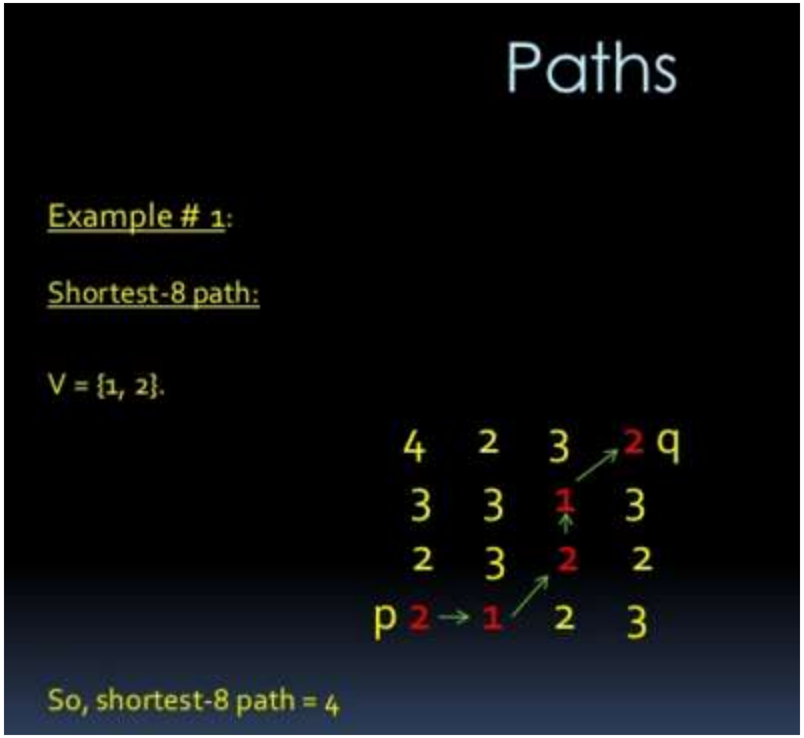
Example # 1:

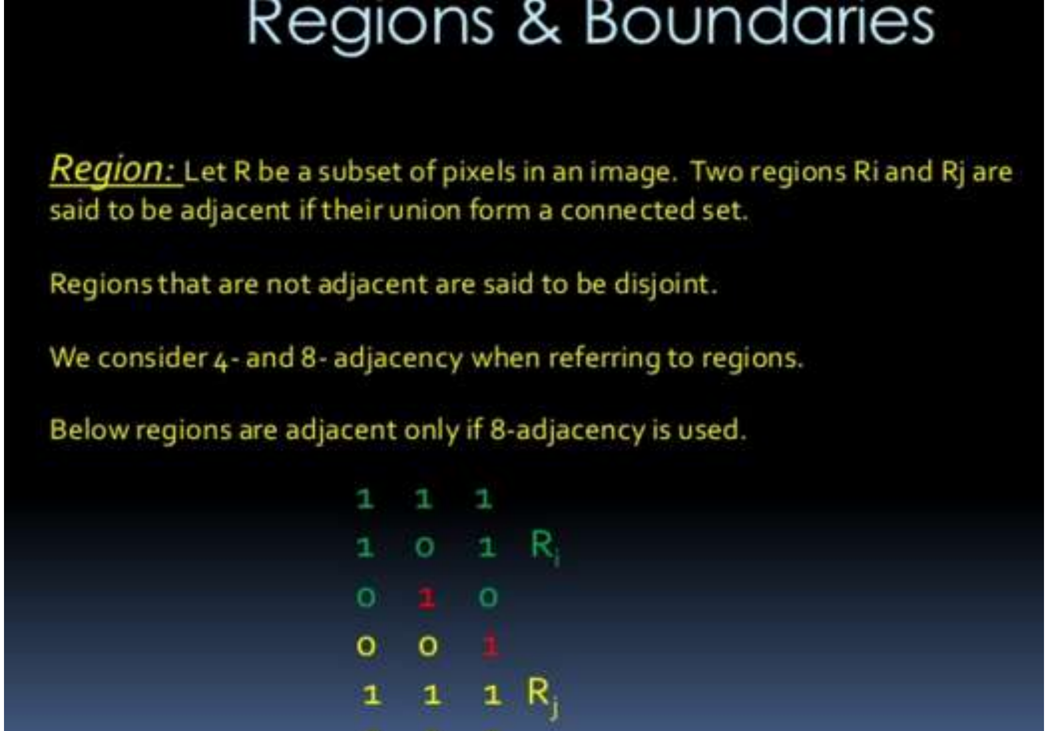
Shortest-8 path:

V = {1, 2}.

	4	2	3	2 q
	3	3	1	3
	2	3	2	2
p	2	1	2	3


Paths
Example # 1:
Shortest-8 path:
V = {1, 2}.
4 2 3 2 q
3 3 1 3
2 3 2 2
p 2 → 1 2 3


Paths
Example # 1:
Shortest-8 path:
V = {1, 2}.
4 2 3 2 q
3 3 1 3
2 3 2 2
p 2 → 1 2 3
So, shortest-8 path = 4



Region

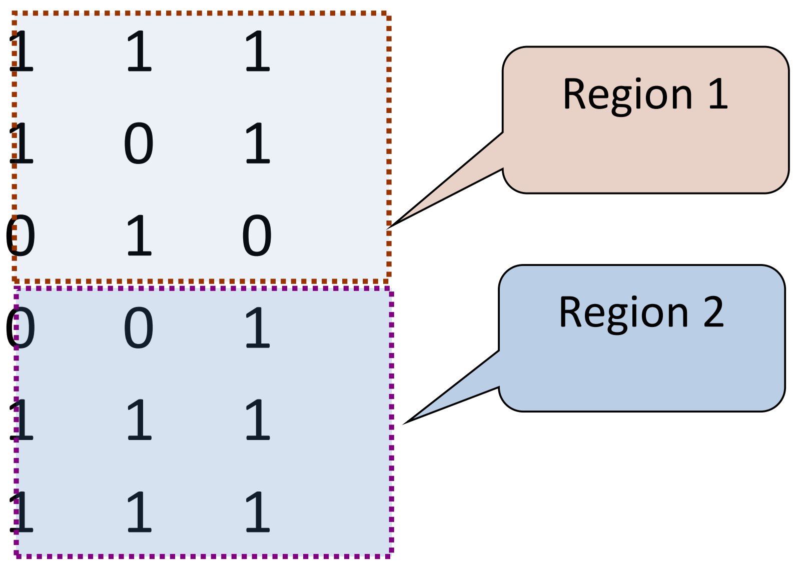
- Let **R** to be a subset of pixels in an image, we call a **R** a region of the image. If **R** is a ***connected*** set.
- Region that are not adjacent are said to be **disjoint**.
- ***Example***: the two regions (of Is) in figure, are adjacent only if 8-adjacany is used.

1	1	1	Ri
1	0	1	
0	1	0	
0	0	1	Rj
1	1	1	
1	1	1	

- ***4-path*** between the two regions does not exist, (so their union in not a connected set).

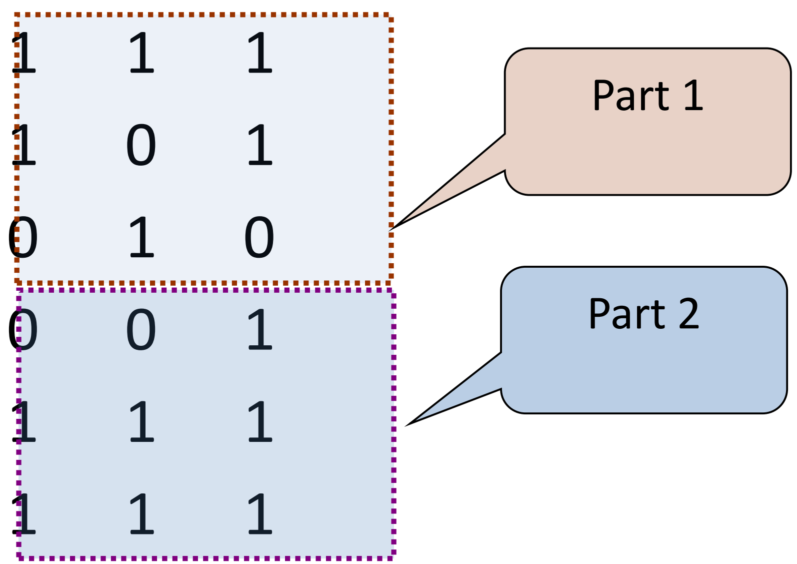
Question 1

- **In the following arrangement of pixels, are the two regions (of 1s) adjacent? (if 8-adjacency is used)**

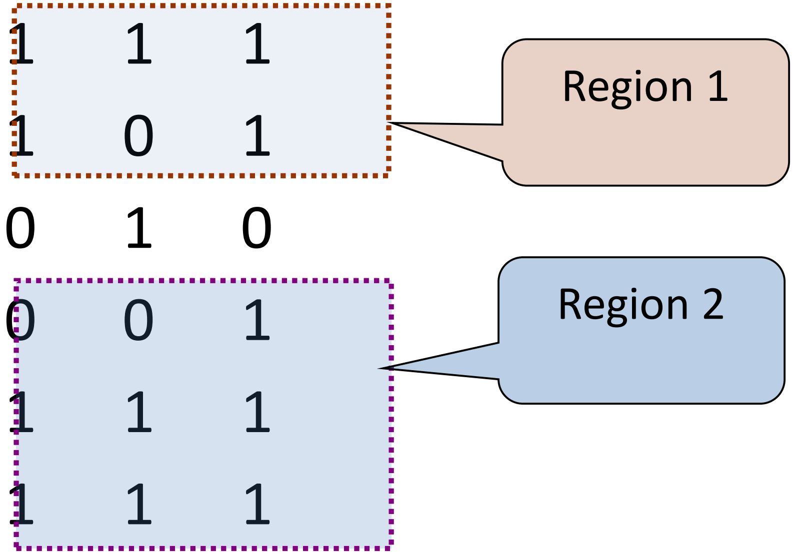


Question 2

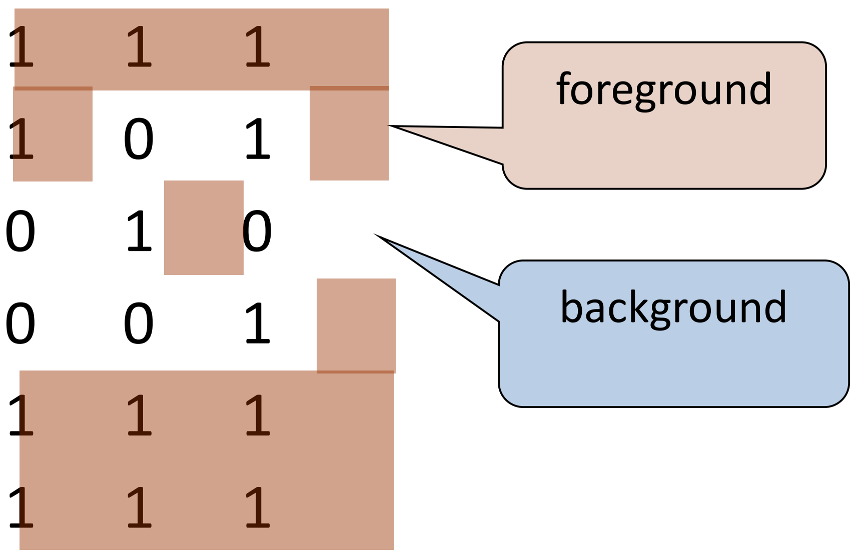
- **In the following arrangement of pixels, are the two parts (of 1s) adjacent? (if 4-adjacency is used)**



- **In the following arrangement of pixels, the two regions (of 1s) are disjoint (if 4-adjacency is used)**



- **In the following arrangement of pixels, the two regions (of 1s) are disjoint (if 4-adjacency is used)**



Question 1

- **In the following arrangement of pixels, are the two regions (of 1s) adjacent? (if 8-adjacency is used)**

1	1	1
1	0	1
0	1	0
0	0	1
1	1	1
1	1	1

Region 1

Region 2

Regions & Boundaries

__Boundaries (border or contour)__: The boundary of a region R is the set of points that are adjacent to points in the compliment of R.

0	0	0	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

RED colored 1 is NOT a member of border if 4-connectivity is used between region and background. It is if 8-connectivity is used.

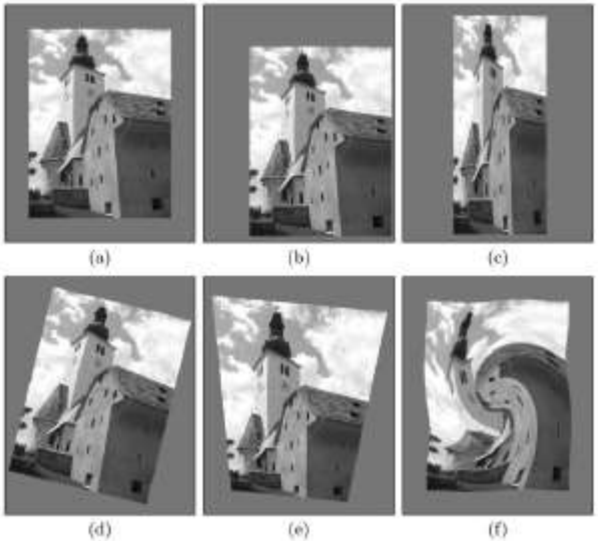
Geometric transformations

Geometric transforms permit the elimination of geometric distortion that occurs when an image is captured.

- An example is an attempt to match remotely sensed images of the same area taken after one year, when the more recent image was probably not taken from precisely the same position.
- To inspect changes over the year, it is necessary first to execute a geometric transformation, and then subtract one image from the other.
- Geometric transformations are widely used for image registration and the removal of geometric distortion. Common applications include construction of mosaics, geographical mapping, stereo and video.

- Geometric operations: change image geometry z Examples: translating, rotating, scaling an image

With geometric transformation, we modify With geometric transformation, we modify the positions of pixels in a image, but keep their colors unchanged – To create special effects – To register two images taken of the same scene at different times – To morph one image to another



Examples of Geometric operations

Geometric Operations

- Example applications of geometric operations:
  - Zooming images, windows to arbitrary size
  - Computer graphics: deform textures and map to arbitrary surfaces
- **Definition:** Geometric operation transforms image *I* to new image *I'* by modifying coordinates of image pixels

$$I(x, y) \rightarrow I'(x', y')$$

- Intensity value originally at (x,y) moved to new position (x',y')

Example: Translation geometric operation moves value at (x,y) to $(x + d_x, y + d_y)$

Simple Mappings

- **Translation:** (shift) by a vector (d_x, d_y)

$$T_x : x' = x + d_x \quad T_y : y' = y + d_y \quad \text{or} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix}$$

- **Scaling:** (contracting or stretching) along x or y axis by a factor s_x or s_y

$$T_x : x' = s_x \cdot x \quad T_y : y' = s_y \cdot y \quad \text{or} \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

- **Rotation:** the image by an angle α

$$T_x : x' = x \cdot \cos\alpha - y \cdot \sin\alpha$$
$$T_y : y' = x \cdot \sin\alpha + y \cdot \cos\alpha$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Companies In this Field In India

- Sarnoff Corporation
- Kritikal Solutions
- National Instruments
- GE Laboratories
- Ittiam, Bangalore
- Interra Systems, Noida
- Yahoo India (Multimedia Searching)
- nVidia Graphics, Pune (have high requirements)
- Microsoft research
- DRDO labs
- ISRO labs

Companies In Bangalore

- Tech point solutions pvt ltd
- Kritical solutions pvt ltd
- Robert bosch engineering and buisness solutions
- Talent HR solutions
- Cyient limited
- Applied materials india private limited
- Skillmine technology
- K20 consulting private limited
- KPIT technologies ltd
- Axcelovate

Transforms

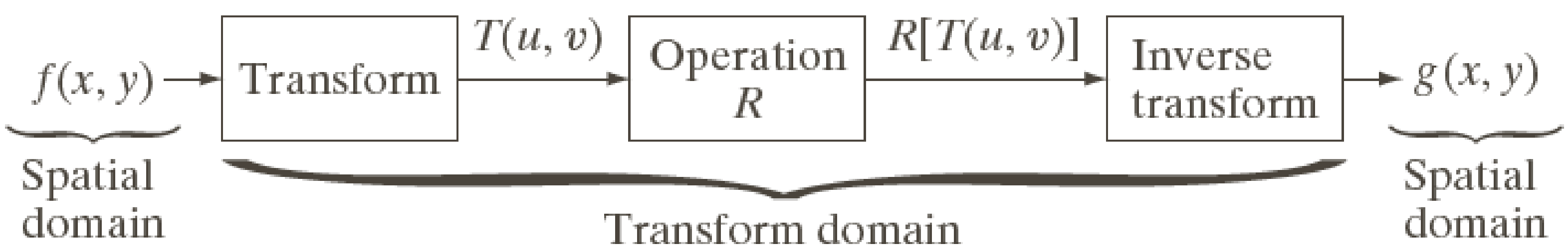
- Example of a substitution:
- Original equation: $x^4 + 4x^2 - 8 = 0$
- Familiar form: $ax^2 + bx + c = 0$
- Let: $y = x^2$
- Solve for y
- $x = \pm\sqrt{y}$
- Transforms are used in mathematics to solve differential equations
- They convert a function from one domain to another with no loss of information

Image Transforms

- Many times, image processing tasks are best performed in a domain other than the *spatial domain.*
- Key steps

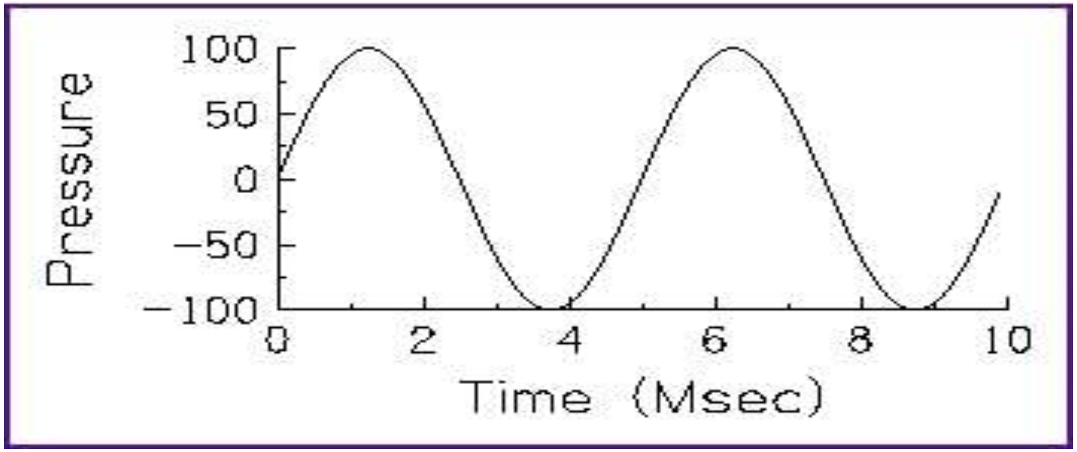
  (1) Transform the image

  (2) Carry the task(s) in the *transformed domain*.



Time Domain and Frequency Domain

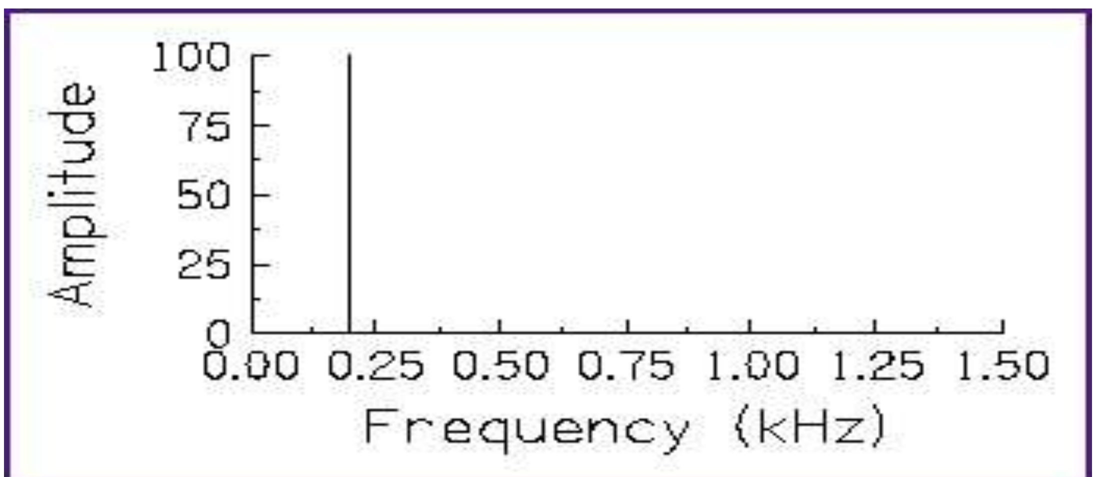
- Time Domain:
  - Tells us how properties (air pressure in a sound function, for example) change over time:



- Am
- Frequency = number of cycles in one second = 200 Hz

Time Domain and Frequency Domain

- Frequency domain:
  - Tells us how properties (amplitudes) change over frequencies:



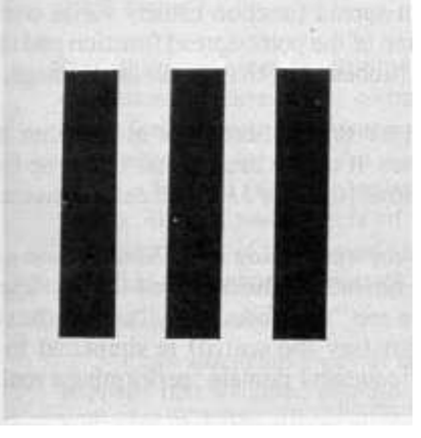
Why is FT Useful?

- Easier to remove undesirable frequencies in the **frequency** domain.

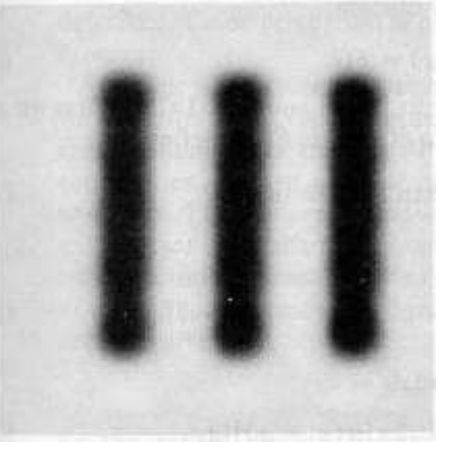
- Faster to perform certain operations in the **frequency** domain than in the **spatial** domain.

How do frequencies show up in an image?

- Low frequencies correspond to slowly varying pixel intensities (e.g., continuous surface).
- High frequencies correspond to quickly varying pixel intensities (e.g., edges)



Original Image



Low-passed

Example of noise reduction using FT

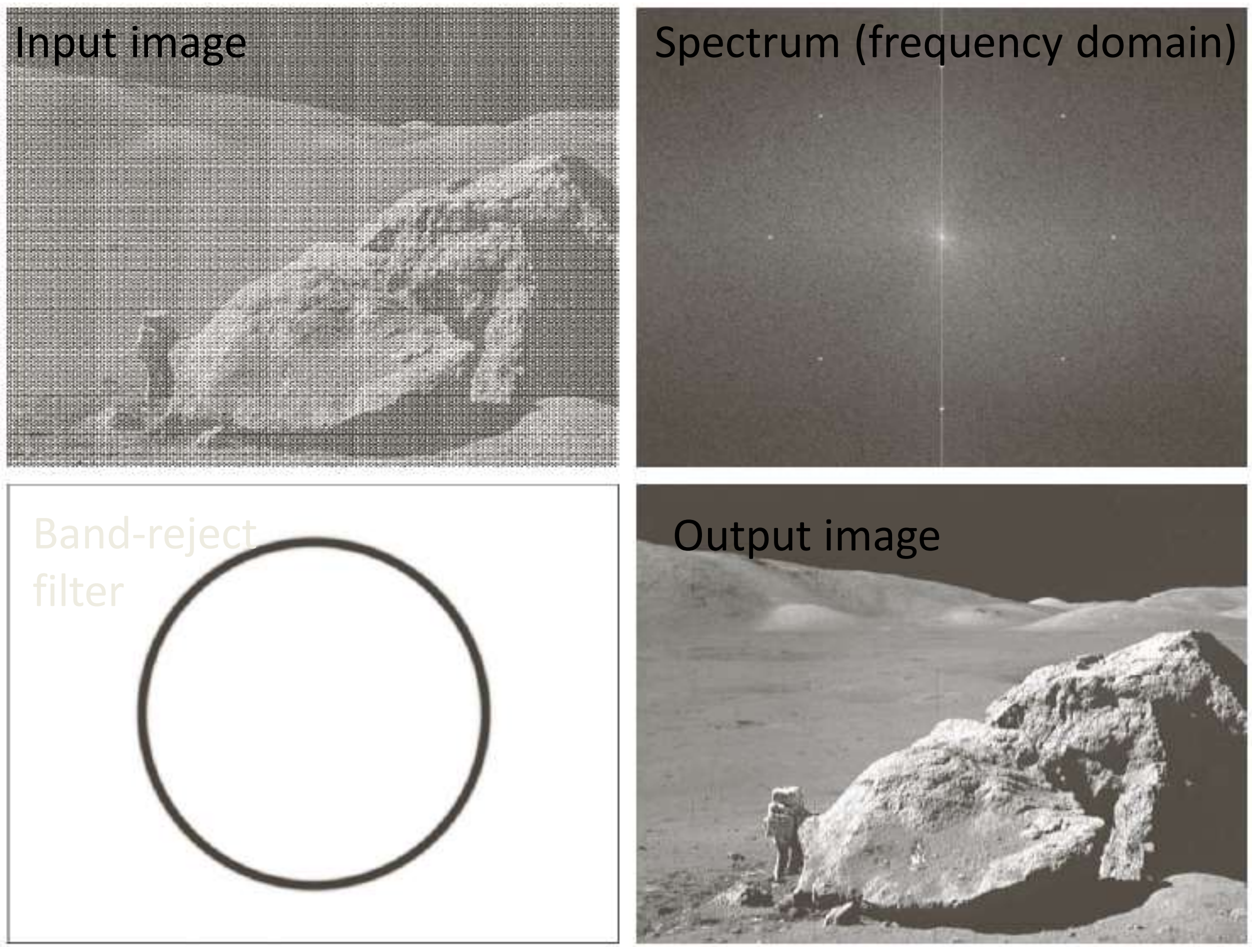
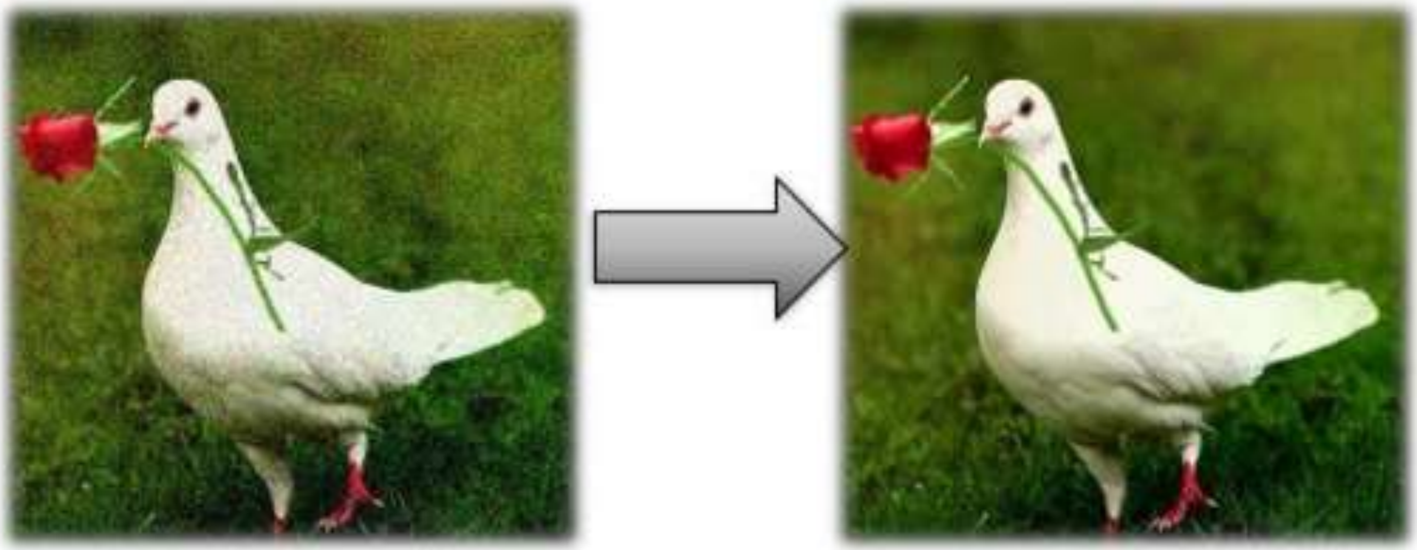


Image Restoration

Digital Image Processing

What is Image Restoration.



Introduction

- **Objective of image restoration**
  - to recover a **distorted image** to the **original form** based on **idealized models**.
- **The distortion is due to**
  - Image degradation in sensing environment e.g. random atmospheric turbulence
  - Noisy degradation from sensor noise.
  - Blurring degradation due to sensors
    - e.g. camera motion or out-of-focus
  - Geometric distortion
    - e.g. earth photos taken by a camera in a satellite

- Image restoration attempts to restore images that have been degraded
  - ✓ Identify the degradation process and attempt to reverse it.
  - ✓ Almost Similar to image enhancement, but more objective.

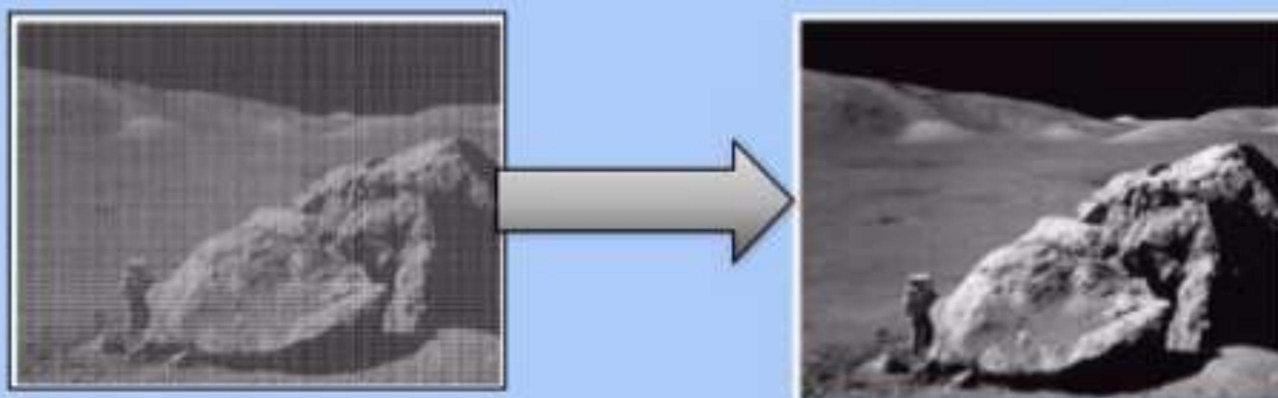
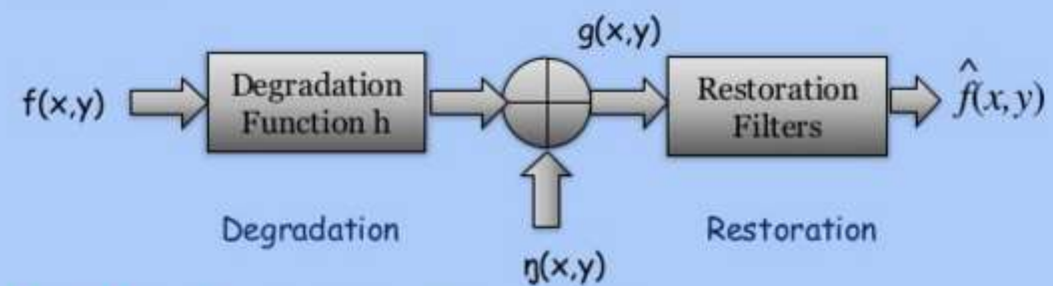


Fig: Degraded image

Fig: Restored image

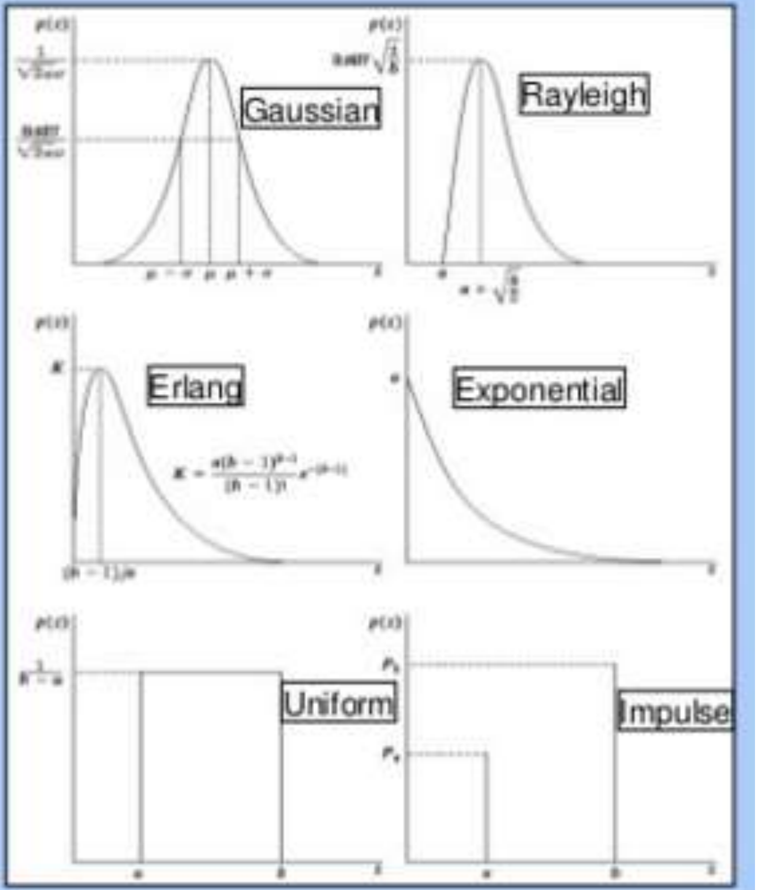
- Image restoration assumes a degradation model that is known or can be estimated.
- Original content and quality does not mean Good looking or appearance.
- Image Enhancement is subjective, where as image restoration is objective process.
- Image restoration try to recover original image from degraded with prior knowledge of degradation process.
- Restoration involves modeling of degradation and applying the inverse process in order to recover the original image.
- Although the restore image is not the original image, its approximation of actual image.

- Objective: To restore a degraded/distorted image to its original conten and quality.



- Spatial Domain: $g(x,y)=h(x,y)*f(x,y)+\eta(x,y)$
- Frequency Domain: $G(u,v)=H(u,v)F(u,v)+\eta(u,v)$
- Matrix: $G=HF+\eta$

- Different models for the image noise term η(x, y)
  - Gaussian
    - Most common model
  - Rayleigh
  - Erlang or Gamma
  - Exponential
  - Uniform
  - Impulse
    - Salt and pepper noise



Noise models

- **Assuming degradation only due to additive noise ($H = 1$)**
- **Noise from sensors**
  - Electronic circuits
  - Light level
  - Sensor temperature
- **Noise from environment**
  - Lightening
  - Atmospheric disturbance
  - Other strong electric/magnetic signals

Noise models

- **Assuming that noise is**
  - independent of spatial coordinates, and
  - uncorrelated with respect to the image content

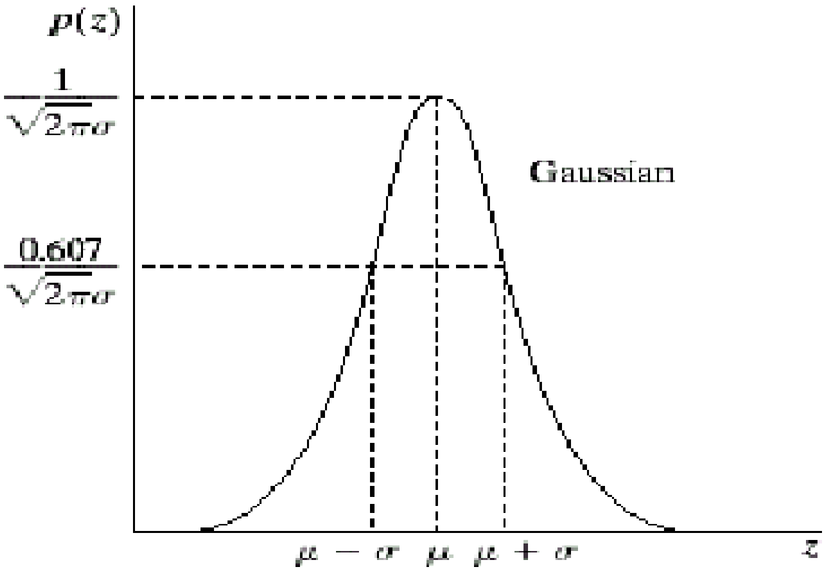
- **Gaussian noise**
  - Probability density function (PDF)

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

  - z: gray level (Gaussian random variable)
  - μ: mean of average value of z
  - σ: standard deviation of z
  - σ^2: variance of z

Noise models

- **PDF of Gaussian noise**
  - 70% of z in $[\mu - \sigma, \mu + \sigma]$
  - 90% of z in $[\mu - 2\sigma, \mu + 2\sigma]$



Noise models

- **Other common noise models**
  - Rayleigh noise
  - Gamma noise
  - Exponential noise
  - Uniform noise

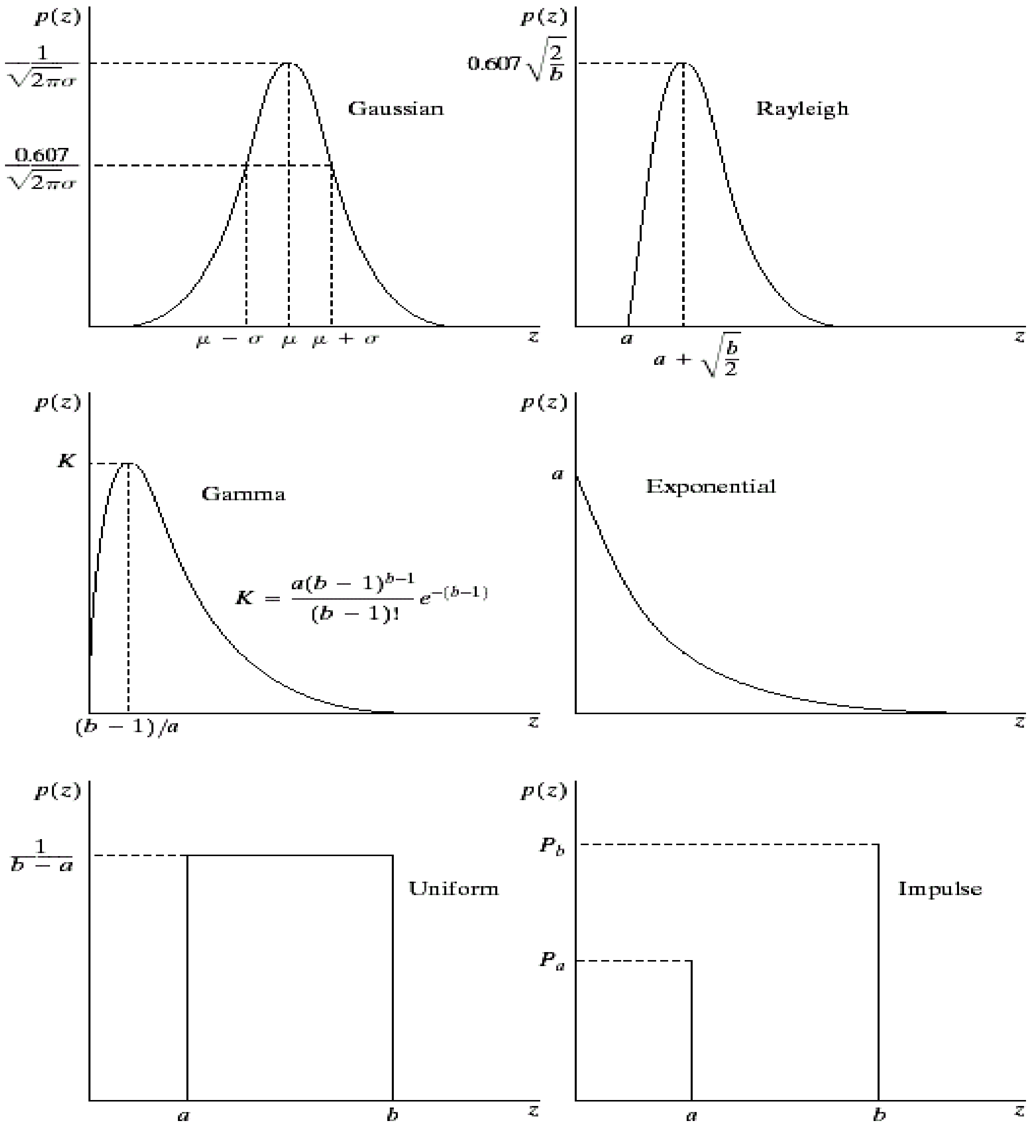
Noise Models

- Rayleigh Noise

$$p(z) = \frac{2}{b}(z-a)e^{-(z-a)^2/b} \quad \text{for } z \ge a$$
$$= 0 \quad \text{for } z < a$$

- Gamma(Erlang) Noise
- Exponential Noise

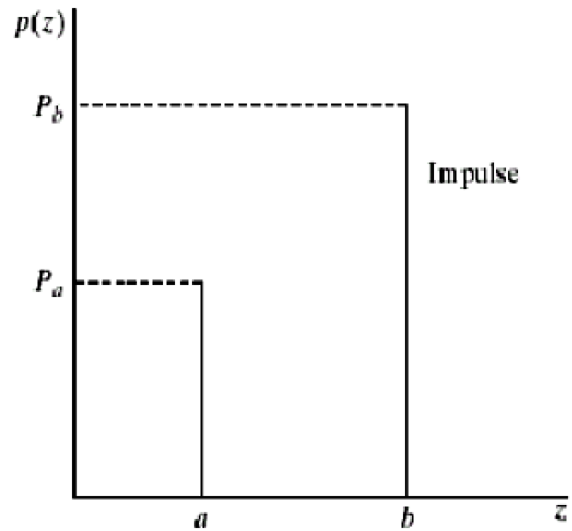
$$p(z) = ae^{-az} \quad \text{for } z \ge 0$$
$$= 0 \quad \text{for } z < 0$$



Noise models

- **Impulse (salt-and-pepper) noise**

$$p(z)=\begin{cases} P_a & \text{for } z=a \\ P_b & \text{for } z=b \\ 0 & \text{otherwise} \end{cases}$$



  - bipolar if $P_a \neq 0,\ P_b \neq 0$
  - unipolar if one of P_a and P_b is 0
  - noise looks like salt-and-pepper granules if $P_a \approx P_b$
  - negative or positive; scaling is often necessary to form digital images
  - extreme values occur (e.g. $a = 0$, $b = 255$)

Applicability of various noise models

- Gaussian noise⇨ electronic circuit noise and sensor noise due to poor illumination and/or high temperature
- Rayleigh density⇨ characterize noise phenomena in range imaging
- Exponential and gamma densities⇨ laser imaging
- Impulse noise⇨ occur when quick transients (faulty switching) take place during imaging
- Uniform density⇨ the least descriptive of practical situations

Estimation of noise parameters

- Periodic noise: Fourier spectral components
- Imaging system is available: study the characteristics of system noise by acquiring a set of images of flat environment under uniform illumination (constant background)
- Only images are available: estimate the noise PDF from small patches of reasonably constant gray level

- Inverse filtering is a deterministic and direct method for image restoration.
- •The images involved must be lexicographically ordered. That means that an image is converted to a column vector by pasting the rows one by one after converting them to columns. •An image of size 256×256 is converted to a column vector of size 65536×1.
- •The degradation model is written in a matrix form, where the images are vectors and the degradation process is a **huge but sparse matrix. g=Hf**
- •The above relationship is ideal. What really happens is **g=Hf+n**!

I

- In this problem we know $\mathbf{H}$ and $\mathbf{g}$ and we are looking for a descent $\mathbf{f}$.

- The problem is formulated as follows:

  We are looking to minimize the Euclidian norm of the error, i.e.,

$$\|\mathbf{n}\|^2 = \|\mathbf{g} - \mathbf{Hf}\|^2$$

- The first derivative of the minimization function must be set to zero.

$$\|\mathbf{g} - \mathbf{Hf}\|^2 = (\mathbf{g} - \mathbf{Hf})^T(\mathbf{g} - \mathbf{Hf}) = \left(\mathbf{g}^T - \mathbf{f}^T\,\mathbf{H}^T\right)(\mathbf{g} - \mathbf{Hf}) =$$

$$\mathbf{g}^T\mathbf{g} - \mathbf{g}^T\,\mathbf{Hf} - \mathbf{f}^T\mathbf{H}^T\,\mathbf{g} + \mathbf{f}^T\mathbf{H}^T\,\mathbf{Hf}$$

$$\frac{\partial\|\mathbf{g}-\mathbf{Hf}\|^2}{\partial\mathbf{f}} = -2\mathbf{H}^T\,\mathbf{g} + 2\mathbf{H}^T\,\mathbf{Hf} = 0 \Rightarrow \mathbf{H}^T\,\mathbf{Hf} = \mathbf{H}^T\,\mathbf{g}$$

$$\mathbf{H}^T\,\mathbf{Hf} = \mathbf{H}^T\,\mathbf{g}$$

$$\mathbf{f} = \left(\mathbf{H}^T\,\mathbf{H}\right)^{-1}\mathbf{H}^T\,\mathbf{g}$$

- If $\mathbf{H}$ is a square matrix and its inverse exists then $\mathbf{f} = \mathbf{H}^{-1}\mathbf{g}$

We have that

$$\mathbf{H}^T \mathbf{H}\mathbf{f} = \mathbf{H}^T \mathbf{g}$$

If we take the DFT of the above relationship in both sides we have:

$$|H(u,v)|^2 F(u,v) = H(u,v)^* G(u,v)$$

$$F(u,v) = \frac{H(u,v)^*}{|H(u,v)|^2} G(u,v)$$

$$F(u,v) = \frac{G(u,v)}{H(u,v)}$$

Note that the most popular types of degradations are low pass filters (out-of-focus blur, motion blur).

- We have that

$$F(u,v) = \frac{G(u,v)}{H(u,v)}$$

- **Problem:** It is very likely that $H(u,v)$ is 0 or very small at certain frequency pairs.

- For example, $H(u,v)$ could be a *sinc* function.

- In general, since $H(u,v)$ is a low pass filter, it is very likely that its values drop off rapidly as the distance of (u,v) from the origin $(0,0)$ increases.

Pseudo-inverse filtering

- Instead of the conventional inverse filter, we implement one of the following:

$$F(u,v) = \begin{cases} \dfrac{G(u,v)}{H(u,v)} & H(u,v) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$F(u,v) = \begin{cases} \dfrac{G(u,v)}{H(u,v)} & |H(u,v)| \geq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

- The parameter ϵ (called **threshold** in the figures in the next slides) is a small number chosen by the user.

- **Enhancement**
  - Concerning the extraction of image features
  - Difficult to **quantify** performance
  - Subjective; making an image “look better”

- **Restoration**
  - Concerning the restoration of degradation
  - Performance can be **quantified**
  - Objective; recovering the original image

Restoration by spatial filtering

- **Assume noise is the only degradation source**

$$g(x,y) = f(x,y) + \eta(x,y)$$

$$G(u,v) = F(u,v) + N(u,v)$$

Noise is unknown

- **Spatial filtering**
  - a means when only additive noise is present
  - similar to enhancement in spatial domain

3×3 filter

Spatial filtering is appropriate when only additive noise is present

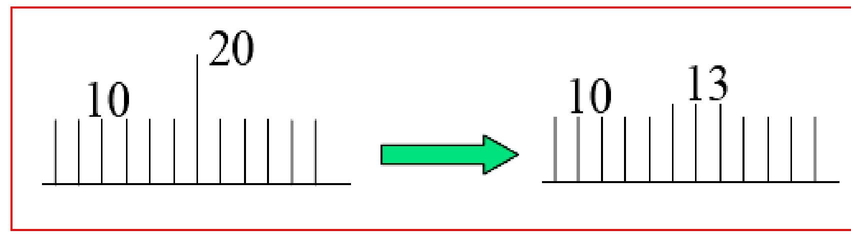
Restoration by spatial filtering

S_{xy} is the set of coordinates in a rectangular subimage window of size $m \times n$ centered at point (x, y)

- **Mean filters (noise reduced by blurring)**
  - **Arithmetic mean filter**

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

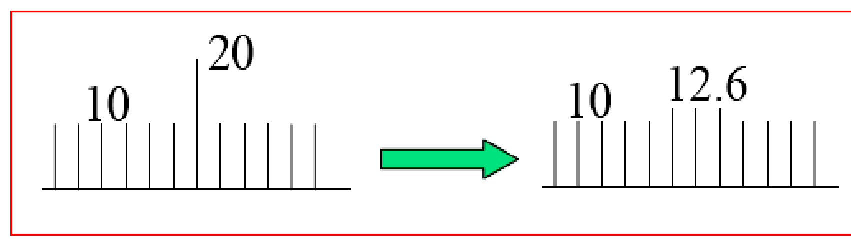
1 x 3 mask



  - **Geometric mean filter**

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

1D illustration



    - Smoothing comparable to arithmetic mean filter
    - Losing less image details

Restoration by spatial filtering

- **An example**
  - Both attenuate noise
  - Image d is sharper than image c

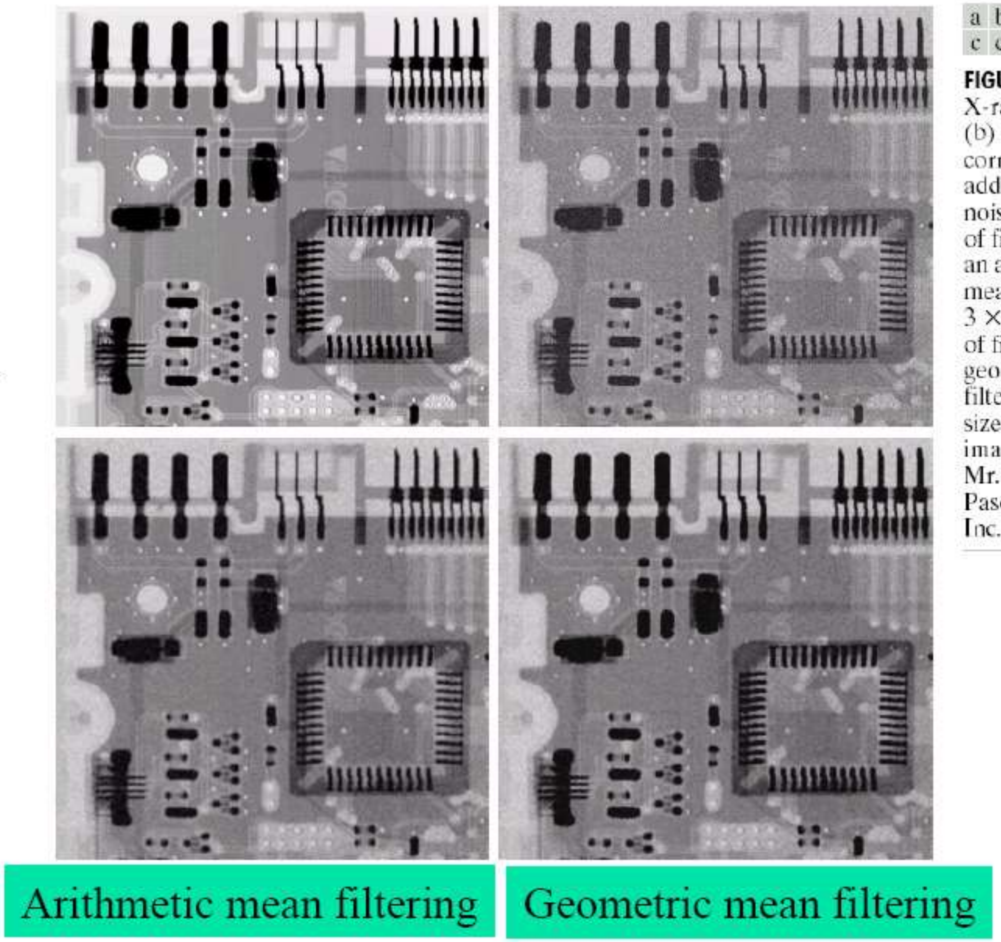
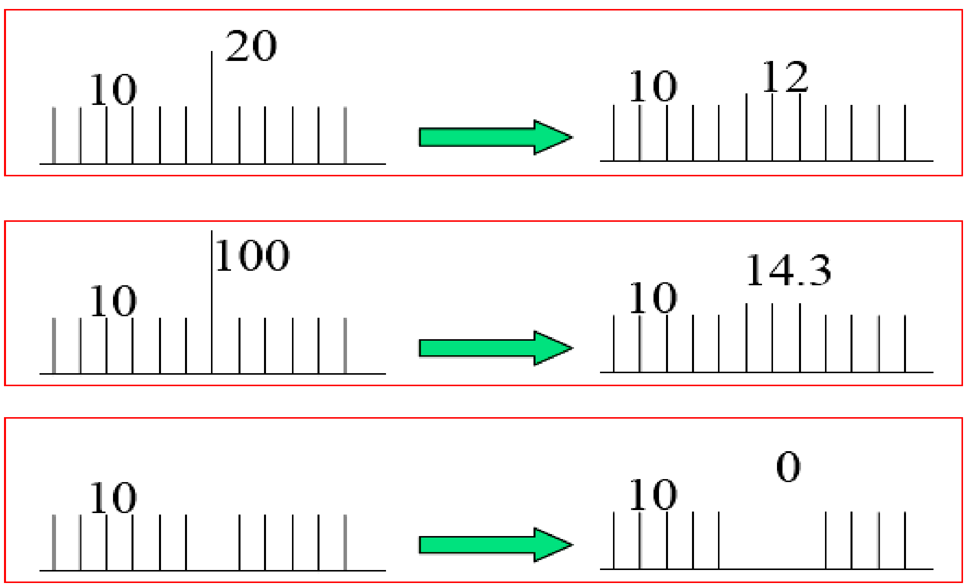


FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Restoration by spatial filtering

- **Mean filters (noise reduced by blurring)**
  - **Harmonic mean filter**
    - good for Gaussian noise
    - good for salt noise
    - bad for pepper noise

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$



good

good

1 x 3 mask

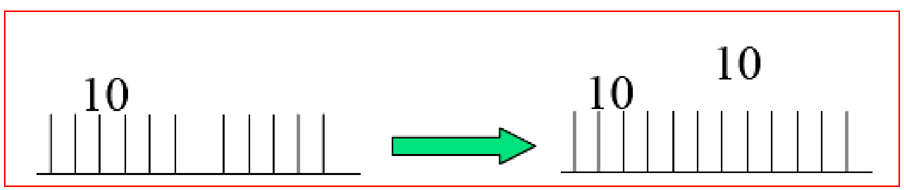
failed

Restoration by spatial filtering

- **Mean filters (noise reduced by blurring)**
  - **Contraharmonic mean filter**
    - $Q > 0$: eliminating pepper noise
    - $Q < 0$: eliminating salt noise
    - cannot do both simultaneously
    - $Q = 0$: arithmetic mean filter
    - $Q = -1$: harmonic mean filter

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Q is the order of filter



$Q = 1$

1 x 3 mask

Restoration by spatial filtering

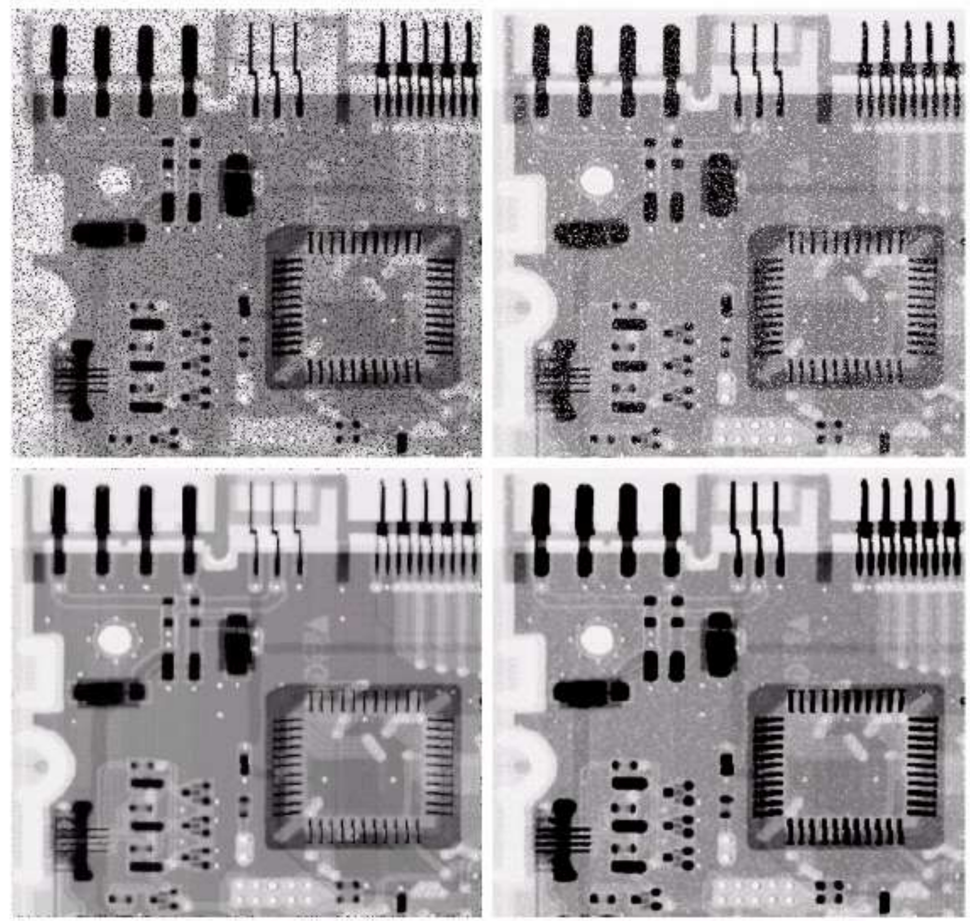
- **An example**
  - good results

Contraharmonic mean filtering

Noise level is
Mean =0
Variance = 400

a	b
c	d

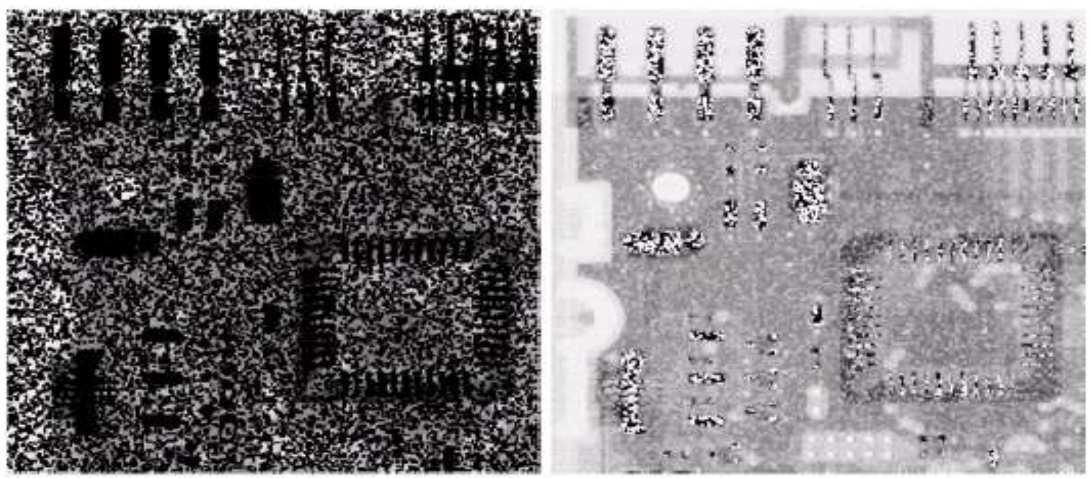
FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3 × 3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.



$Q = 1.5$ $Q = -1.5$

Restoration by spatial filtering

- **Mean filters (noise reduced by blurring)**
  - Arithmetic mean filter and geometric mean filter are well suited for random noise such as Gaussian noise
  - Contraharmonic mean filter is well suited for impulse noise
    - Disadvantage: must know pepper noise or salt noise in advance



a b

FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3×3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

Disastrous results

Restoration by spatial filtering

- **Order-statistics filters**
  - **Median filter**
    - handling both bipolar or unipolar impulse noise

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s,t)\}$$

  - **Max filter**
    - finding the brightest points in an image
    - reducing pepper noise

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}}\{g(s,t)\}$$

  - **Min filter**
    - finding the darkest points in an image
    - reducing salt noise

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}}\{g(s,t)\}$$

Restoration by spatial filtering

- **Order-statistics filters**
  - **Midpoint filter**
    - order statistics
    - averaging
    - work well for Gaussian noise

$$\hat{f}(x,y)=\frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\}+\min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

  - **Alpha-trimmed mean filter**
    - $d=0$: arithmetic mean filter
    - $d=mn-1$: median filter
    - suitable for the situation involving multiple types of noise

$$\hat{f}(x,y)=\frac{1}{mn-d}\sum_{(s,t)\in S_{xy}} g_r(s,t)$$

delete d/2 lowest and d/2 highest values first, then average the remaining

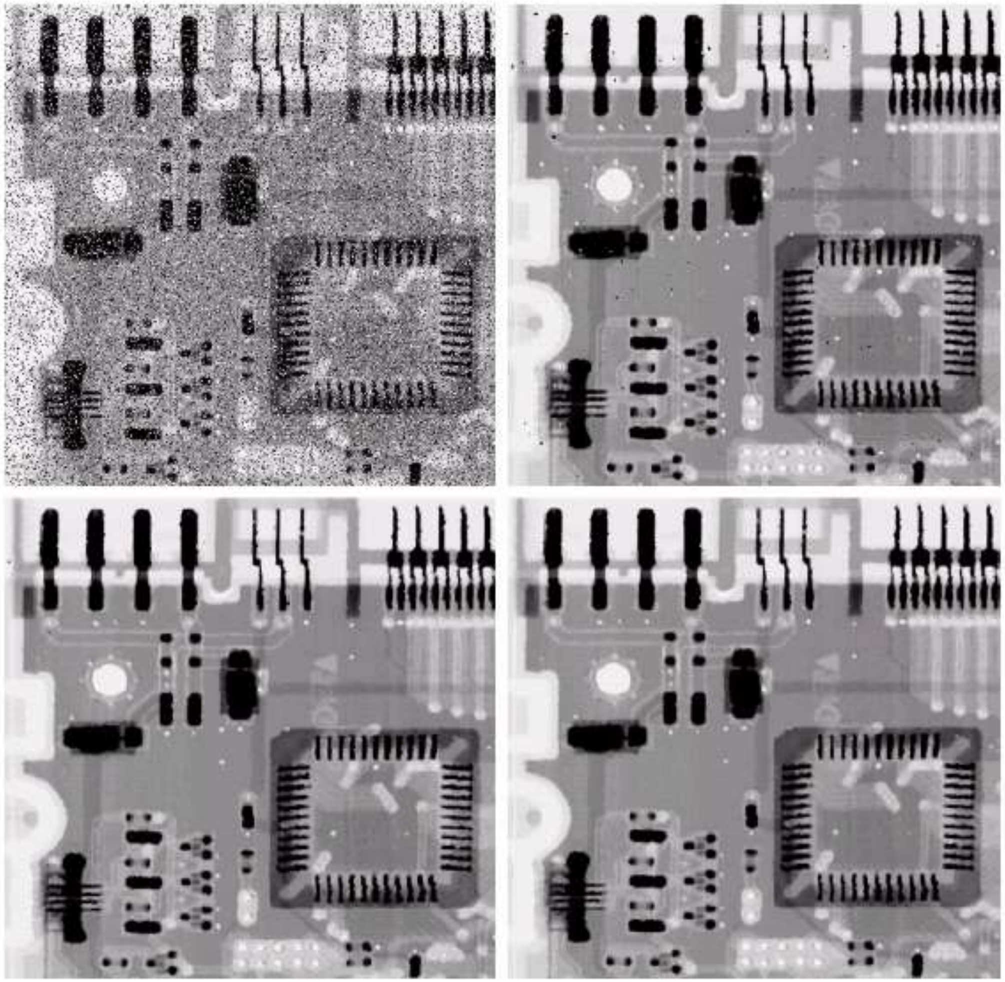
wrong

Restoration by spatial filtering

- An example

a b
c d

FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

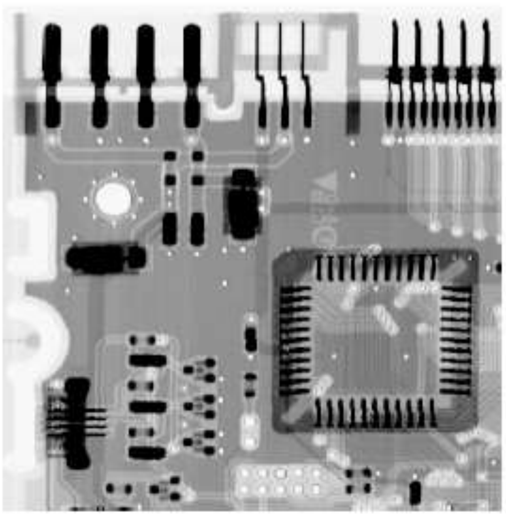


Median filtering

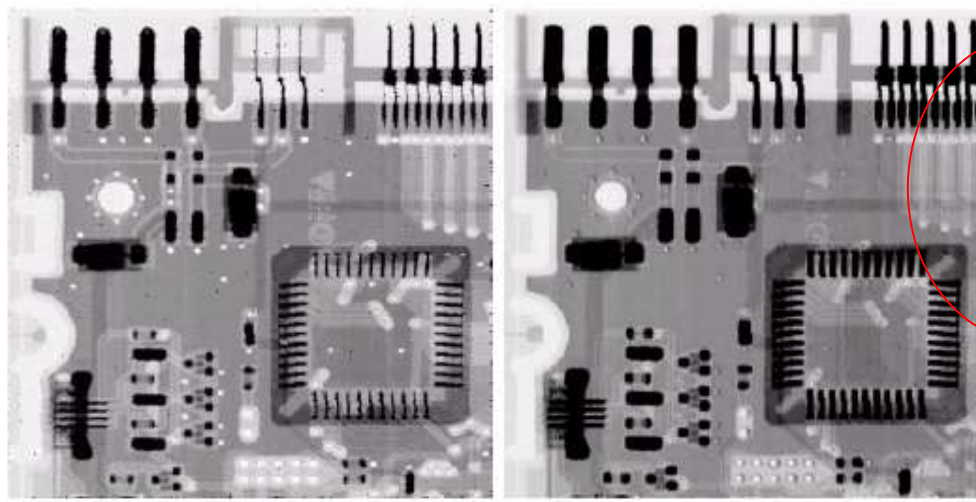
-- Repeated passes of median filter tend to blur the image.
-- Keep the number of passes as low as possible.

Restoration by spatial filtering

- **An example**



Original image



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

Fig. 8 next page

Max filtering removes pepper noise

It also removes some dark pixels

Min filtering remove salt noise

It also removes some white pixels

Restoration by spatial filtering

Pepper noise

Salt noise

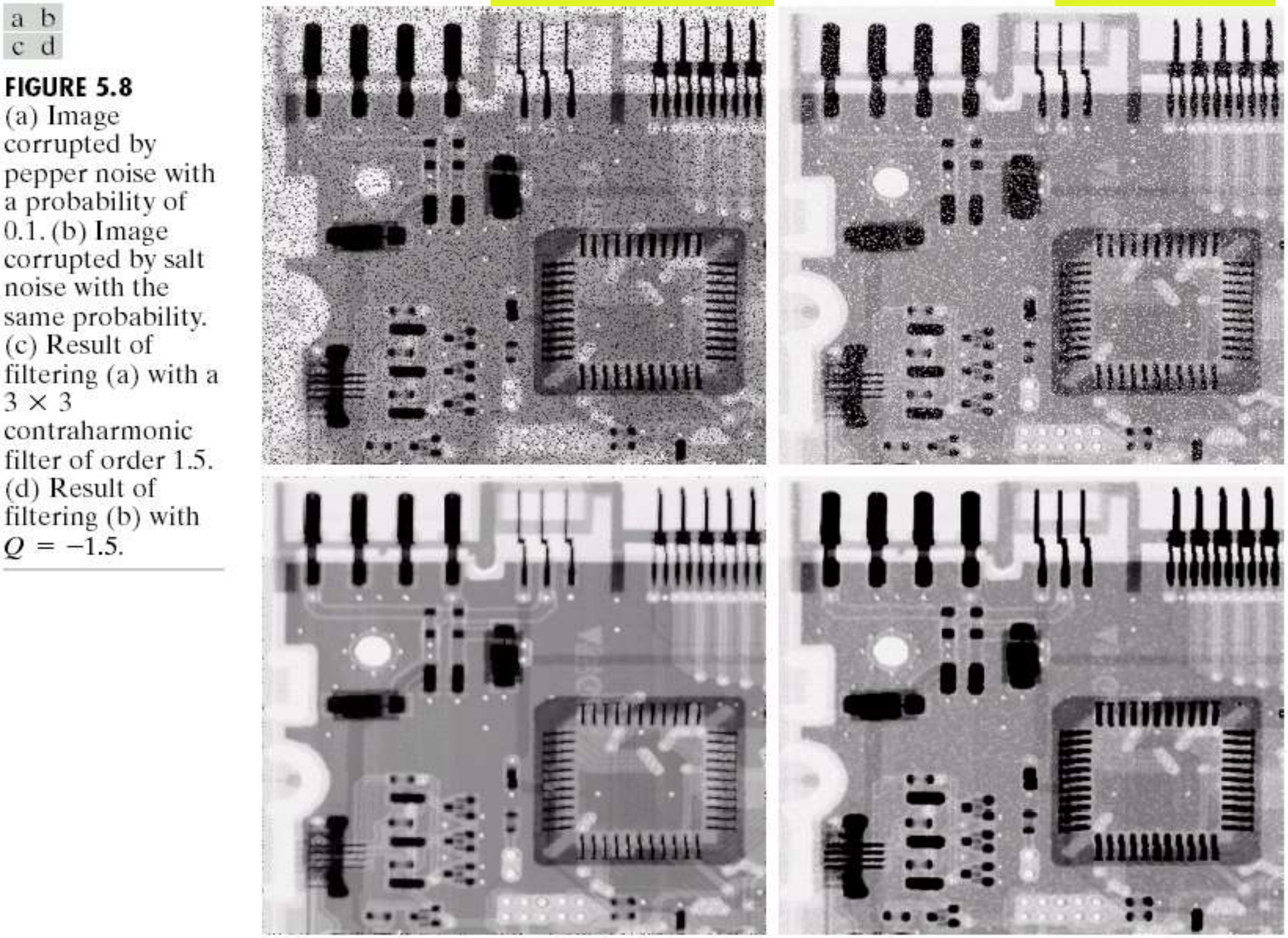
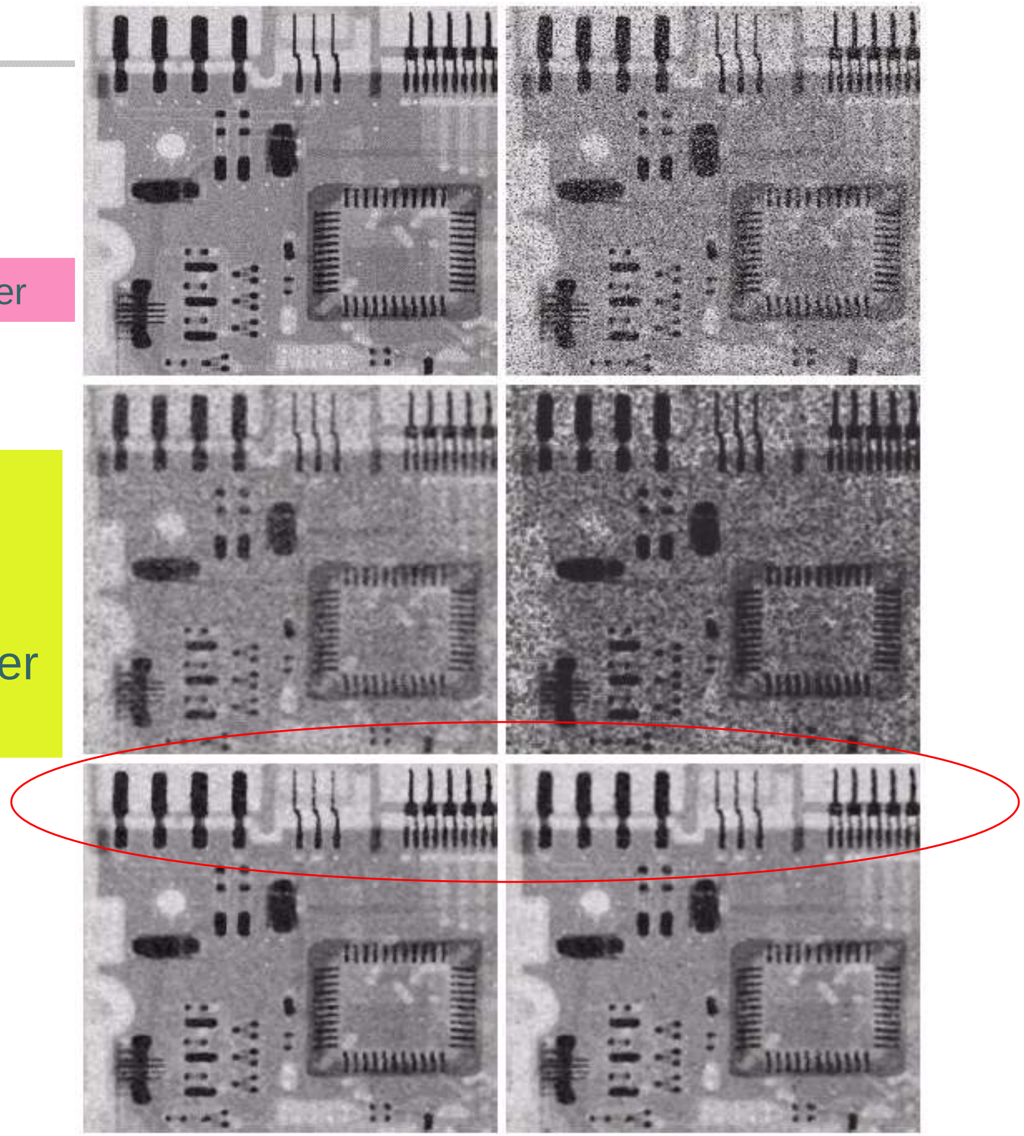


FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3 × 3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a 5×5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.



-High level of noise ➔ large filter

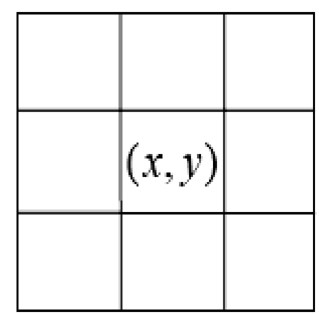
-Median and alpha-trimmed filter performed better
- Alpha-trimmed did better than median filter

Restoration by spatial filtering

- **Filters discussed so far**
  - Do not consider image characteristics
- **Adaptive filters to be discussed**
  - Behaviors based on statistical characteristics of the subimage under a filter window
  - Better performance
  - More complicated
  - **Adaptive, local noise reduction filter**
  - **Adaptive median filter**

Restoration by spatial filtering

- **Adaptive, local noise reduction filter**
  - **Mean** of a random variable: a measure of average gray level in some region
  - **Variance** of a random variable: a measure of average contrast in the region
  - Response based on four quantities
    - $g(x,y)$: value of noisy image at (x,y)
    - σ_η^2 : variance of the noise
    - m_L : local mean in S_{xy}
    - σ_L^2 : local variance in S_{xy}



S_{xy}

Restoration by spatial filtering

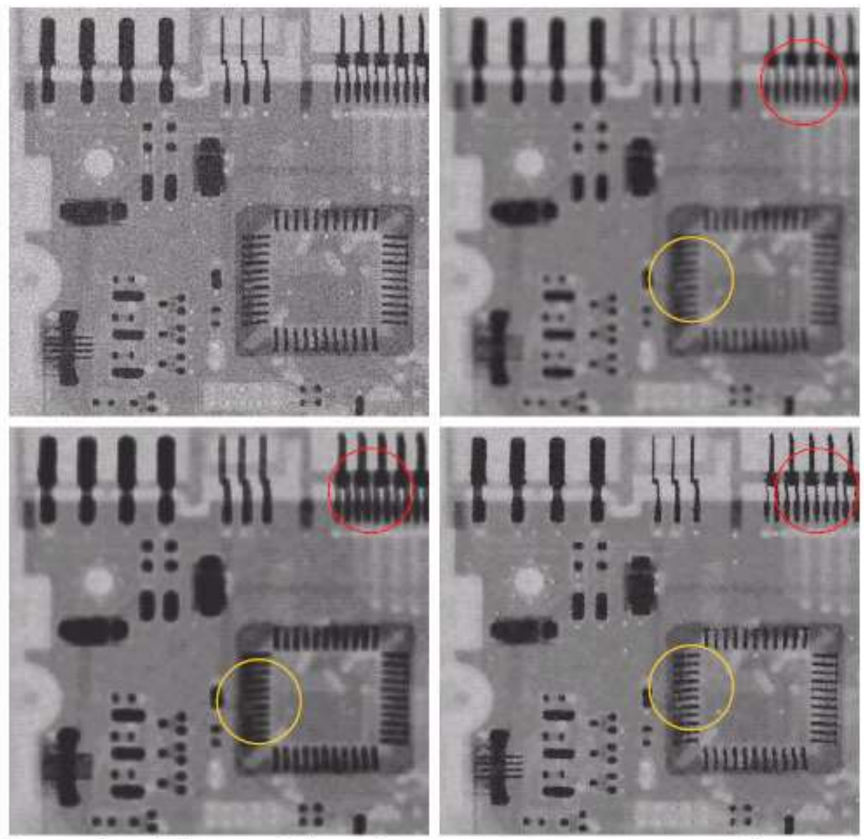
- **An example of results obtained by three filters**

$\sigma_\eta^2 = 1000$

Filter size
7 x 7

a b
c d

FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7 × 7.



Noise reduced but significant blurring

Adaptive filtering

Noise reduced but significant blurring

Best result

Restoration by spatial filtering

- **Adaptive median filter**
  - Objective:
    - Removing salt-and-pepper noise
    - Reducing distortion (excessive thinning/thickening of object boundaries)
  - Size of S_{xy} changes during filtering operation

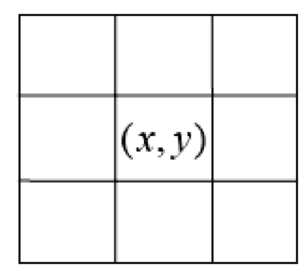
$$z_{\text{min}} = \text{ minimum gray level in } S_{xy}$$

$$z_{\text{max}} = \text{ maximum gray level in } S_{xy}$$

$$z_{\text{med}} = \text{ median of gray levels in } S_{xy}$$

$$z_{xy} = \text{ gray level at } (x, y)$$

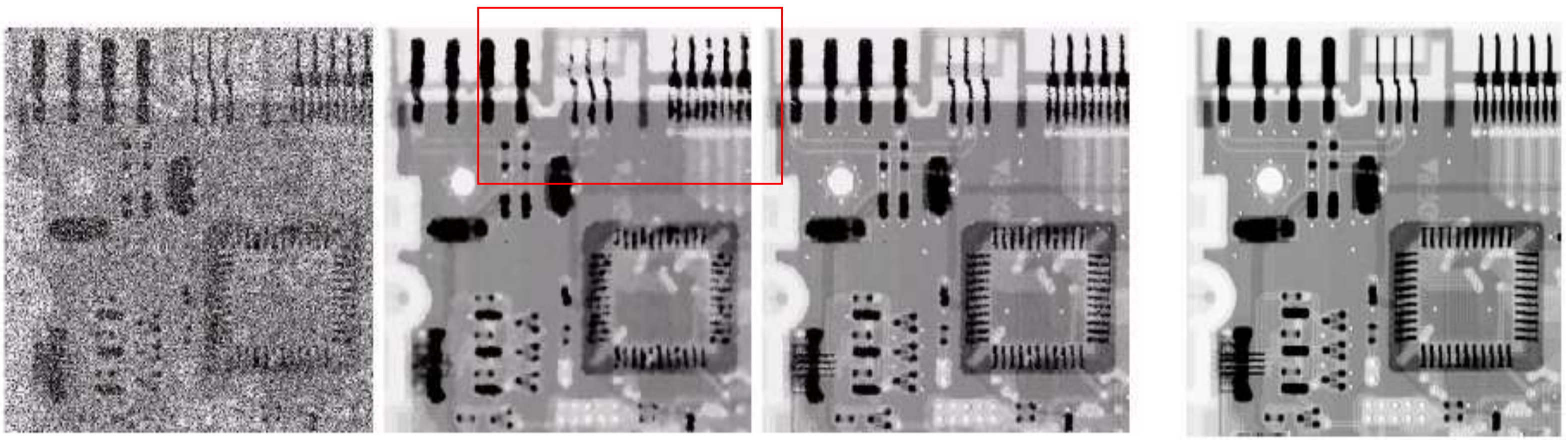
$$S_{\text{max}} = \text{ maximum allowed size of } S_{xy}$$



S_{xy}

Restoration by spatial filtering

- An example of results by two filters



a b c

7×7 $S_{max} = 7 \times 7$ Original image

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

Noise removed effectively but significant loss of details

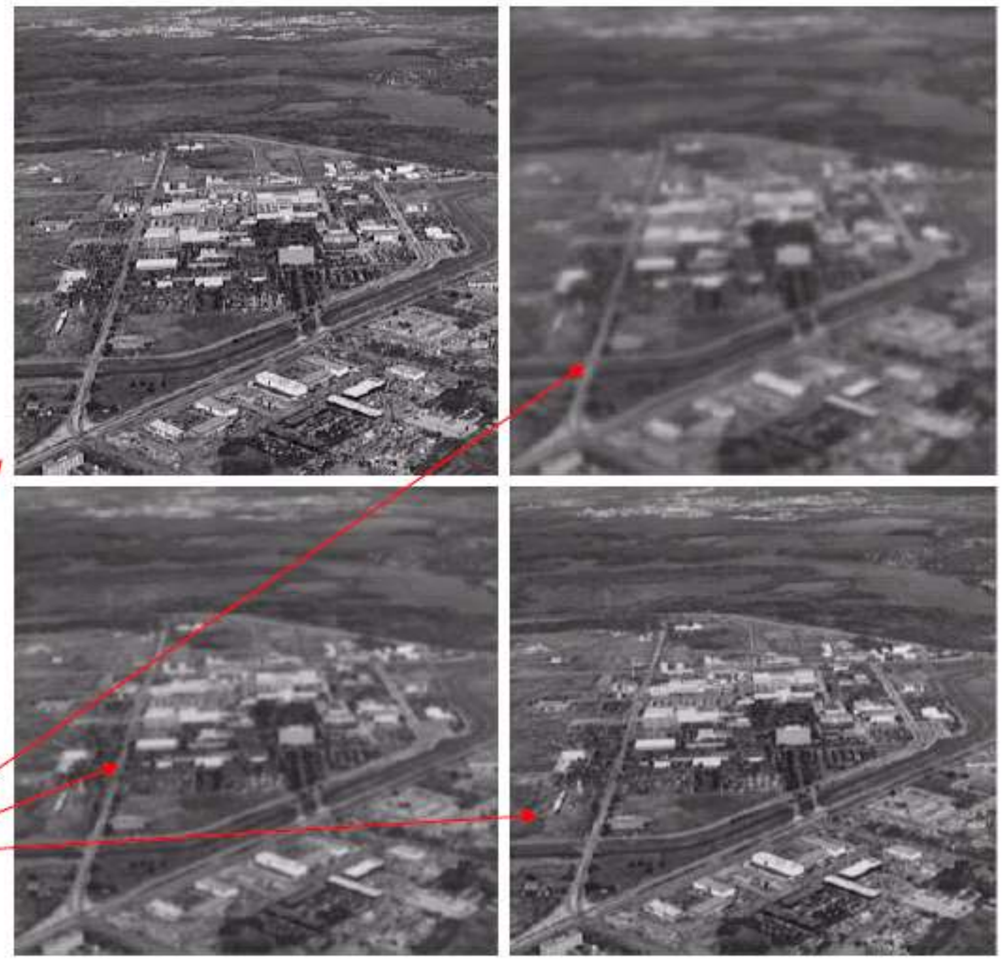
Noise removed effectively and preserving details and sharpness

Estimation of degradation functions

An example

a b
c d

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)



$$f(x,y)$$

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}}$$

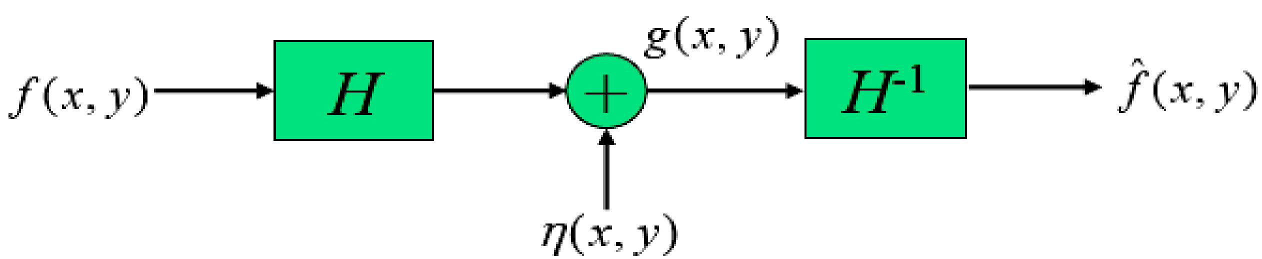
$$H(u,v)F(u,v) \Leftrightarrow g(x,y)$$

Inverse filtering

- **When $H(u,v)$ is known, the simplest approach to restoration is direct inverse filtering**

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$



Inverse filtering

- However, even if H is known completely, the undegraded image cannot be recovered exactly due to noise N

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

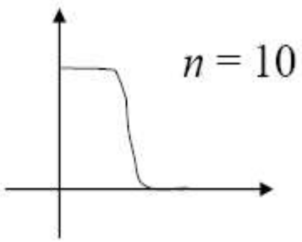
- Even worse when H has zero or very small values, N/H would dominate the estimated image
- One way to get around this problem is to limit the filter frequencies to values near the origin where H is large in general

Inverse filtering

Degradation function $H(u,v) = e^{-k\left[(u-M/2)^2+(v-N/2)^2\right]^{5/6}}$

- **An example**

Butterworth filter H_b



$$\frac{G(u,v)}{H(u,v)} = \hat{F}(u,v) \Leftrightarrow \hat{f}(x,y)$$

$$\frac{G(u,v)}{H(u,v)} H_b(u,v) = \hat{F}(u,v)$$

$$\Updownarrow$$

$$\hat{f}(x,y)$$

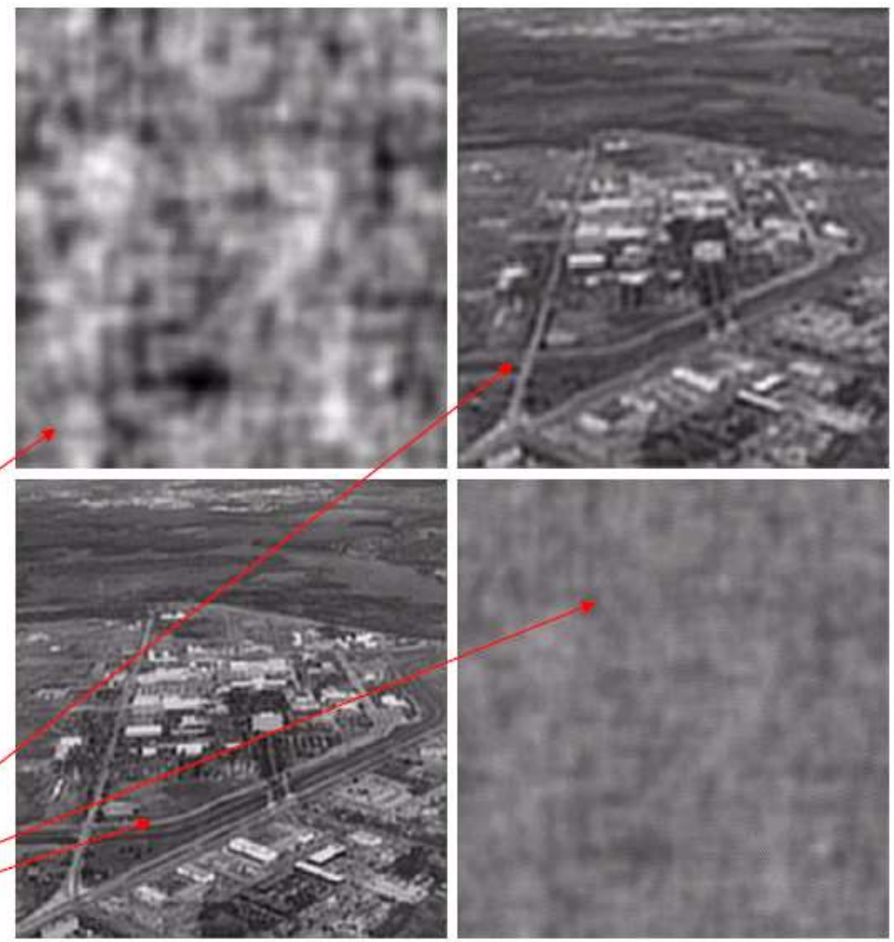


FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with H cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.

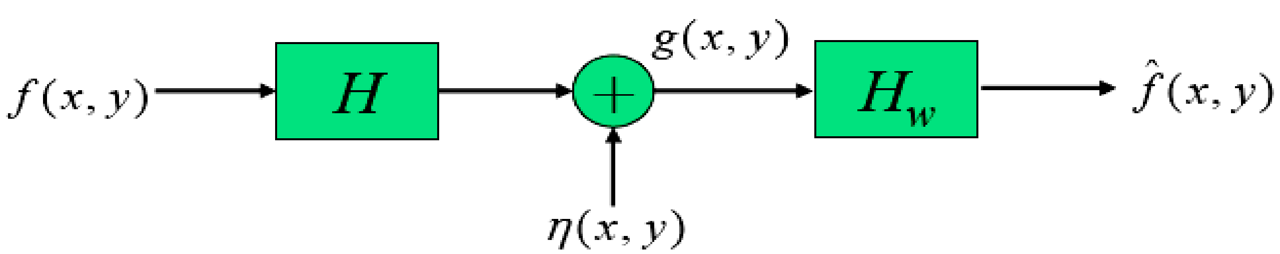
Curtain of noise

Cutting off values of the ratio outside a radius of 40, 70,85.

Wiener filtering

- **Main limitation of inverse filtering**
  - Very sensitive to noise
- **Wiener filtering (minimum mean square error filtering)**
  - Use **statistic information** about signal and noise to improve the restoration
  - Consider images and noise as **random processes**
  - Objective:

$$e^2 = E\{(f - \hat{f})^2\} \Rightarrow \min$$



Wiener filtering

- **Assumptions**
  - The noise and the image are uncorrelated
  - One or the other has zero mean
- **Estimated image**

$$\min\{e^2\} \Rightarrow \hat{F}(u,v)$$

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)$$

$$|H(u,v)|^2 = H^*(u,v)H(u,v)$$

$H(u,v)$ is the degradation function

$$S_\eta(u,v) = |N(u,v)|^2 : \text{power spectrum of the noise}$$

$$S_f(u,v) = |F(u,v)|^2 : \text{power spectrum of the original image}$$

$$(\eta(x,y) \Leftrightarrow N(u,v),\ f(x,y) \Leftrightarrow F(u,v))$$

Wiener filtering

- **Wiener filter**

$$H_w(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}$$

- **When noise is zero** $(N(u,v) = 0)$

$$H_w(u,v) = \frac{1}{H(u,v)} \Rightarrow \text{the inverse filter}$$

- **When** $S_\eta(u,v)$ **and/or** $S_f(u,v)$ **are unknown**

White noinse

$$H_w(u,v) \approx \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}$$

some constant

Wiener filtering

- **An example**

$$\frac{G(u,v)}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+K}=\hat{F}(u,v)\Leftrightarrow\hat{f}(x,y)$$

$$\frac{G(u,v)}{H(u,v)}H_b(u,v)=\hat{F}(u,v)\Leftrightarrow\hat{f}(x,y)$$

$$H(u,v)=e^{-k(u^2+v^2)^{5/6}}$$

$$\frac{G(u,v)}{H(u,v)}=\hat{F}(u,v)\Leftrightarrow\hat{f}(x,y)$$

original image f degraded image $f*h$

a b c

FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.